

APP:PLE PI <나만의 TODO앱 만들기> 클래스

소프트웨어 나눔축제

강의자 문태서, 최재혁



01. 소개합니다

애플파이와 오늘 강의 내용을 알아봅시다



App : ple Pi 는요...

선린 **최초**의 모바일 어플리케이션 & 창업 동아리

교내 대회 수상만 **12개** (2021년)

2021 소프트웨어과 **최우수** 동아리 선정

이미 위의 문장으로 설명이 끝난 것 같지만
더 많은 정보가 알고 싶다면 **applepi.com**



〈나만의 TODO앱 만들기〉 강의를 소개합니다



애플파이 12기 최재혁

1교시 소개 + 미니게임

오늘의 강의 내용을
살펴보는 시간입니다

4교시 Kotlin 알아보기

투두앱의 코드와 함께
코틀린을 알아봅니다

〈나만의 TODO앱 만들기〉 강의를 소개합니다



2교시 XML 알아보기

XML이 무엇인지
코드로 배워봅니다

3교시 XML 실습하기

배운 XML을 활용해
실습을 진행합니다

애플파이 13기 문태서

이제 여러분들을 소개 할 차례입니다!

첫 번째 사람이 자기소개 하면,
다음 사람은 자기소개를 이어나가는
미니게임 자기소개



1번 학생
“나는 문태서야” (이름)



2번 학생
“나는 최재혁이야” (이름)
“내 MBTI는 ENTJ야” (MBTI)



마지막 학생
이름, MBTI 부터 시작해서
자신이 원하는 것까지 자기소개

주제로 무엇이 나왔을까?

모르거나, 답변하기 어려운 질문은 넘기면 됩니다 :)

이름 ->

#01. 미니게임

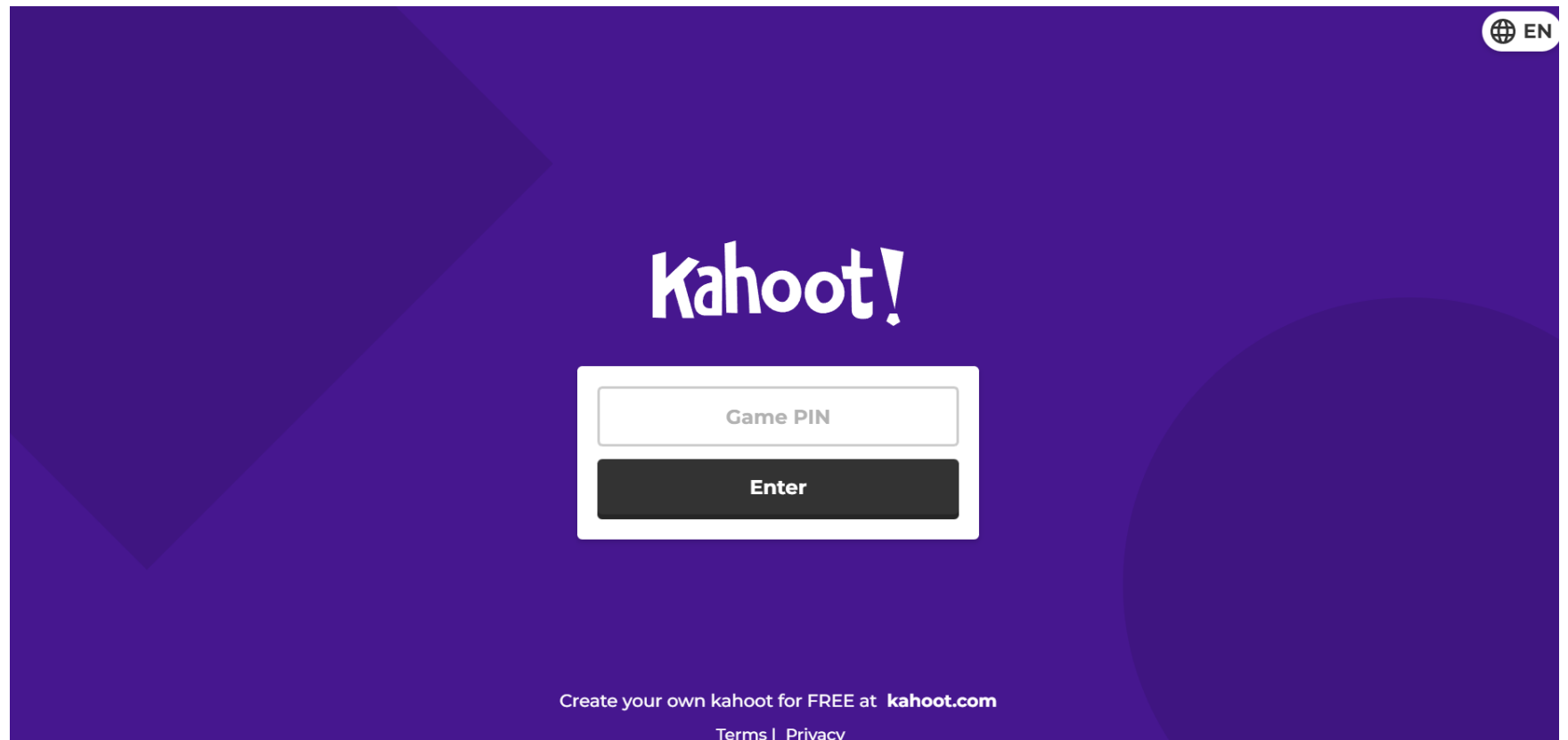
#02. XML소개

#03. XML실습

#04. Kotlin

여러분들의 개성 있는 별명으로 아래에 접속하세요!

<https://kahoot.it/>





#4교시. 코틀린

TODO앱의 코틀린 코드를 알아봅시다

#깃허브 프로젝트 클론하기

아래 주소를 주소창에 입력하여 깃허브에 접속합니다.
https://bit.ly/2022_SSF_GIT

The screenshot shows the GitHub interface for the repository 'App-ple-Pi / SSFProjectBlank'. The repository is public and has 0 stars, 0 forks, and 0 watches. The main branch is 'main'. The repository contains several files and folders, including '.idea', 'app', 'gradle/wrapper', '.gitattributes', '.gitignore', 'build.gradle', 'gradle.properties', and 'gradlew'. The repository was created 2 days ago and has 2 commits. The repository description is '2022 애플파이 소프트웨어 나눔축제'.

Repository: App-ple-Pi / SSFProjectBlank (Public)

Buttons: Edit Pins, Watch (0), Fork (0), Star (0)

Navigation: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings

Branches: main (1 branch), 0 tags

Buttons: Go to file, Add file, Code

Commit: jaehyeok3017 Minor: MainActivity.kt 086cfe1 2 days ago 2 commits

File/Folder	Commit	Time
.idea	Minor: MainActivity.kt	2 days ago
app	Minor: MainActivity.kt	2 days ago
gradle/wrapper	Initial commit	2 days ago
.gitattributes	Initial commit	2 days ago
.gitignore	Initial commit	2 days ago
build.gradle	Initial commit	2 days ago
gradle.properties	Initial commit	2 days ago
gradlew	Initial commit	2 days ago

About: 2022 애플파이 소프트웨어 나눔축제

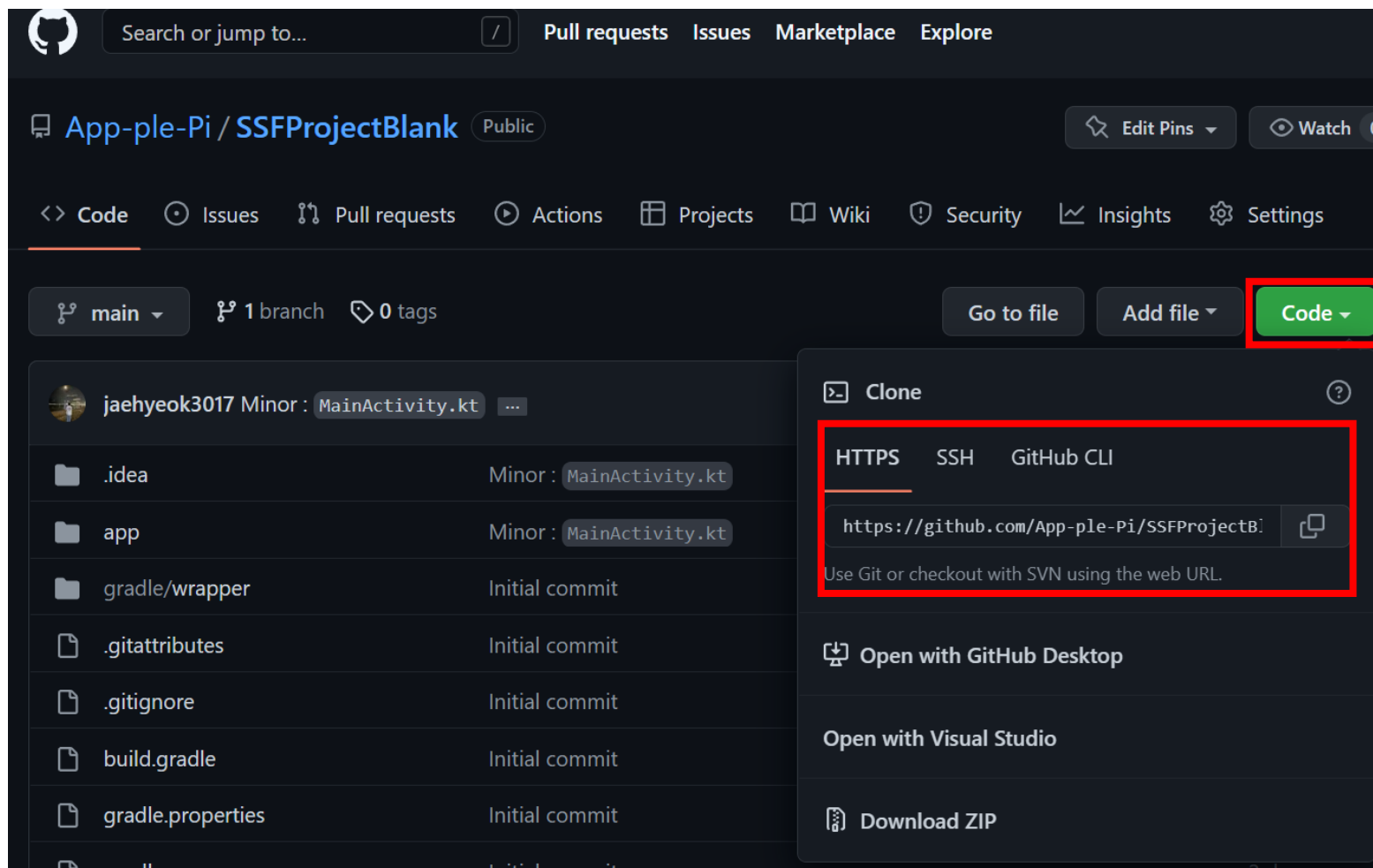
0 stars, 0 watching, 0 forks

Releases: No releases published. Create a new release

Packages: No packages published

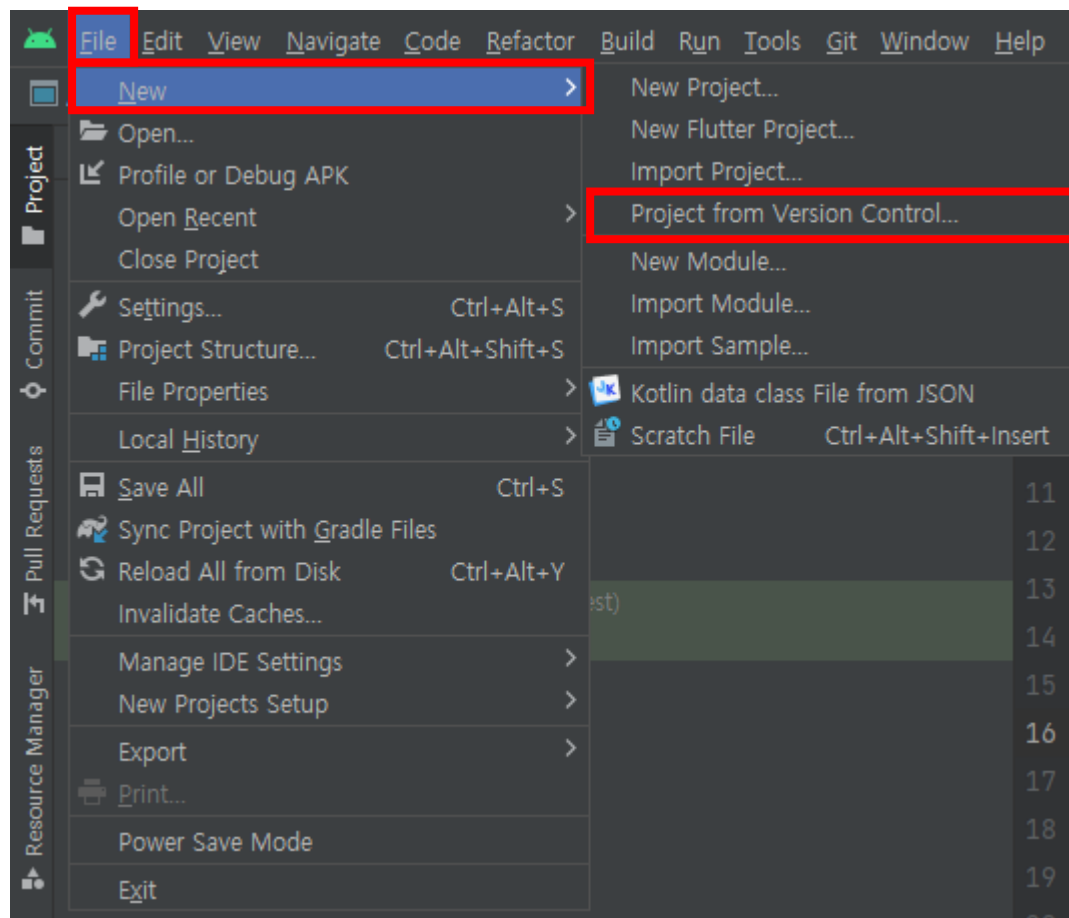
#깃허브 프로젝트 클론하기

아래 초록색 Code 버튼을 누른 뒤,
주소 옆 오른쪽 네모 두 개 버튼을 눌러 링크를 복사합니다.



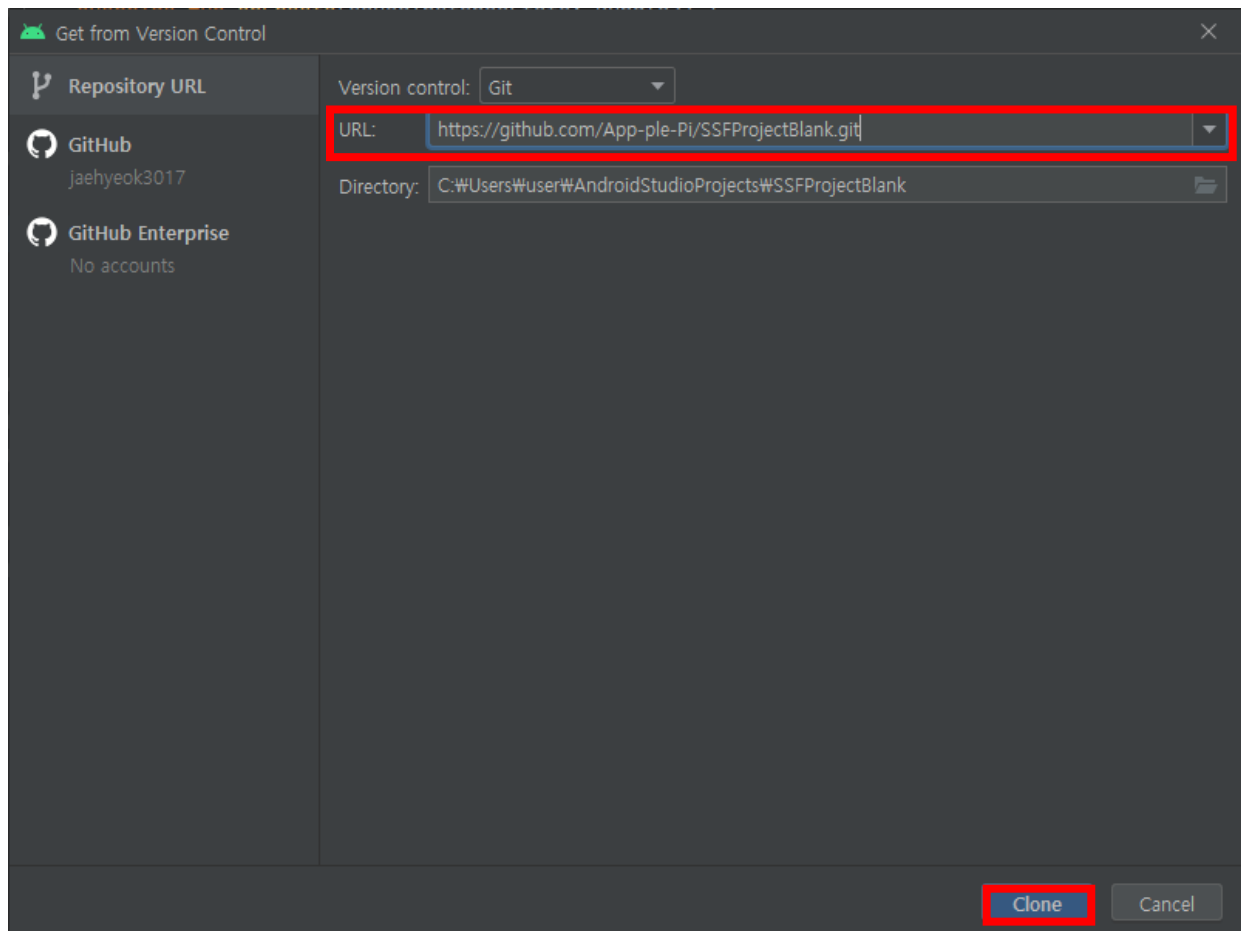
#깃허브 프로젝트 클론하기

**안드로이드 스튜디오를 접속한 뒤,
File -> New -> Project from Version Control을 눌러줍니다**



#깃허브 프로젝트 클론하기

**URL : 빈칸으로 되어있는 칸에
복사한 주소를 붙여 넣어 주고, Clone 버튼을 눌러줍니다.**



#깃허브 프로젝트 클론하기

깃허브 프로젝트 클론 PPT를 다시 보려면
아래 주소를 접속해서 볼 수 있습니다

https://bit.ly/2022_SSF_DOCS

#01. 미니게임

#02. XML소개


#03. XML실습

#04. Kotlin

XML

TextView, ImageView,
EditText, Button

ID 지정, Margin 지정,
Width, height 지정,
Constraint 지정
(ex : Top_toTopOf)


AchieveList

우선순위 : ○ 1 ○ 2 ○ 3

Item 0

Item 1

Item 2

Item 3

Item 4

Item 5

Item 6

Item 7

Item 8

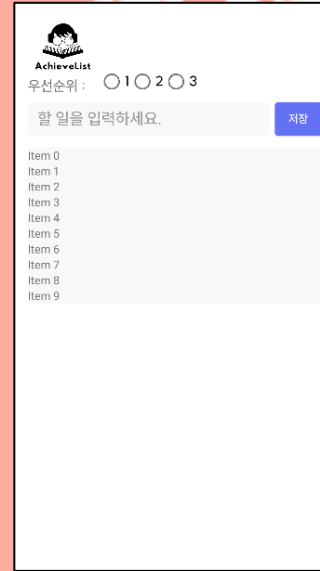
Item 9

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin



앱 개발

XML
레이아웃


Kotlin
기능

#01. 미니게임

#02. XML소개

#03. XML실습


#04. Kotlin


AchieveList
우선순위 : ○ 1 ○ 2 ○ 3

plz sleep

저장

라디오 버튼 클릭 후,
→
저장 버튼을 누를 시


AchieveList
우선순위 : ○ 1 ○ 2 ● 3

plz sleep

저장

☐ plz sleep

×

☐ plz sleep

×

☐ plz sleep

×

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

XML

TextView, ImageView,
EditText, Button

ID 지정, Margin 지정,
Width, height 지정,
Constraint 지정
(ex : Top_toTopOf)

Kotlin

Button을 눌렀을 때,
TODO를 저장시키는 것

SplashActivity(초기 화면)에서
MainActivity로
화면을 전환하는 것

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

Kotlin

Button을 눌렀을 때,
TODO를 저장시키는 것

SplashActivity(초기 화면)에서
MainActivity로
화면을 전환하는 것

Button을 눌렀을 때?
어떤 버튼인데?
라디오 버튼도 버튼 아닌가?

**버튼이 무엇인지
정확하게 지정할 필요** 있음

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin



변경여부
변경여부

변수명 : 자료형 = 값 할당
변수명 : 자료형

값 할당을 할 경우
자료형을 적을 필요가 없음

변수 선언만 할 경우
자료형을 반드시 적어야 함

가변 - var (Changeable)	값을 계속 할당이 가능함
불변 - val (Unchangeable)	값을 한 번만 할당이 가능함 (값을 재할당하려고 했을 시, 에러)

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin



변경여부 변수명 : 자료형 = 값 할당

1. 카멜 표기법

(ex. camelCase)

각 단어의 첫 문자를 대문자로 적는다
단, 가장 첫 번째 글자는 소문자로 적는다

2. 스네이크 표기법

(ex. snake_case)

모든 단어를 소문자로 사용하고,
각 단어 사이에 _ (언더바)를 넣어서 적는다

3. 파스칼 표기법

(ex. PascalCase)

카멜 표기법과 같지만,
첫 번째 글자까지 대문자를 사용해서 적는다

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin



변경여부 변수명 : 자료형 = 값 할당

추후 변수를 바꿀 것인지?

가변 - var Changeable	값을 계속 할당이 가능함
불변 - val Unchangeable	값을 한 번만 할당이 가능함 (값을 재할당하려고 했을 시, 에러)

```
fun main() {  
    var a = 5  
    a = 3  
}
```

a 값은 정상적으로 3으로 변경됨
(에러 없이 정상적으로 변경)

```
fun main() {  
    val b = 5  
    b = 3  
}
```

Val cannot be reassigned
Val은 재할당이 불가하다는 오류 발생

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin



변경여부 변수명 : 자료형 = 값 할당

형태	이름	크기 (sizeof)	사용 예시
정수형	Byte	1 byte	var num1 : Byte = 123
	short	2 byte	var num2 : Short = 123
	Int (기본)	4 byte	var num3 = 123 (10진수를 표기하려면 0x를, 2진수를 표기하려면 0b를 앞에 붙이기)
	long	8 byte	var num4 : Long = 123L (L을 통해 더 큰 메모리를 가진 변수임을 알려줘야 함)
실수형	float	4 byte	var num5 : Float = 123.5f
	double (기본)	8 byte	var num6 = 123.4
문자형	Char (한 개의 문자)		var str1 = 'a'
문자열	String (한 개 이상의 문자)		var str2 = "String"
논리형	Boolean		var bool = true

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin



변경여부 변수명 : 자료형 = 값 할당

Q. 자료형이 일치하지 않다면?

```
fun main() {  
    var num1 : Int = 123  
    var num2 : Double = num1  
}
```

Type mismatch: inferred type is Int but Double was expected
(유형이 일치하지 않음 : 추론된 형식은 Int이지만, 더블 형식이 예상됨)

Q. 자료형을 변경하고 싶다면?

```
fun main() {  
    var num1 : Int = 123  
    var num2 : Double = num1.toDouble()  
}
```

.to자료형 을 이용하여, 자료형을 변환할 수 있음
(.toInt(), .toDouble(), .toByte(), .toFloat(), toChar() 등)

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin



```
val addBtn = findViewById<Button>(R.id.mainToDoButton)
```



변경여부 변수명 : 자료형 = 값 할당



가변 - var (Changeable)	값을 계속 할당이 가능함
불변 - val (Unchangeable)	값을 한 번만 할당이 가능함 (값을 재할당하려고 했을 시, 에러)



findViewById?
값이 무엇이
할당 된 걸까?

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin



MainActivity.kt

```
val addBtn = findViewById<Button>(R.id.mainT0DoButton)
```

XML

activity_main.xml

```
<Button
    android:id="@+id/mainT0DoButton"
    android:layout_width="58dp"
    android:layout_height="42dp"
    android:layout_marginTop="94dp"
    android:layout_marginRight="16dp"
    android:layout_marginBottom="14dp"
    android:background="@drawable/storage_button_round"
    android:fontFamily="@font/suitmedium"
    android:text="저장"
    android:textColor="#FFFFFF"
    android:textStyle="normal"
    app:layout_constraintBottom_toTopOf="@id/mainToDoRecyclerView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@id/mainToD0Edittext" />
```

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin



MainActivity.kt

XML

activity_main.xml

findViewById<View>(R.id.ID)

XML(레이아웃)에서 선언한 View = Button
설정한 id = mainToDoButton

```
val addBtn = findViewById<Button>(R.id.mainToDoButton)
```



```
<Button  
    android:id="@+id/mainToDoButton"  
    android:layout_width="58dp"  
    android:layout_height="42dp"  
    android:layout_marginTop="94dp"  
    android:layout_marginRight="16dp"  
    android:layout_marginBottom="14dp"  
    android:background="@drawable/storage_button_round"  
    android:fontFamily="@font/suitmedium"  
    android:text="저장"  
    android:textColor="#FFFFFF"  
    android:textStyle="normal"  
    app:layout_constraintBottom_toTopOf="@id/mainToDoRecyclerView"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toEndOf="@id/mainToDoEditText" />
```

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

findViewById<View>(R.id.ID)

XML(레이아웃)에서 선언한 View = Button
설정한 id = mainToDoButton

위를 이용하여 TODO 5와 6을 해결 해봅시다!

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

Answer.
TODO 5

findViewById<View>(R.id.ID)

XML(레이아웃)에서 선언한 View = Button
설정한 id = mainToDoButton

```
val addBtn = findViewById<Button>(R.id.mainToDoButton)
```

```
<Button
    android:id="@+id/mainToDoButton"
    android:layout_width="58dp"
    android:layout_height="42dp"
    android:layout_marginTop="94dp"
    android:layout_marginRight="16dp"
    android:layout_marginBottom="14dp"
    android:background="@drawable/storage_button_round"
    android:fontFamily="@font/suitmedium"
    android:text="저장"
    android:textColor="#FFFFFF"
    android:textStyle="normal"
    app:layout_constraintBottom_toTopOf="@id/mainToDoRecyclerView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@id/mainToDoEditText" />
```


#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

Answer.
TODO 6

findViewById<View>(R.id.ID)

XML(레이아웃)에서 선언한 View = Button
설정한 id = mainToDoButton

```
var addEditText = findViewById<EditText>(R.id.mainToDoEditText)

<EditText
    android:id="@+id/mainToDoEditText"
    android:layout_width="300dp"
    android:layout_height="42dp"
    android:layout_marginStart="24dp"
    android:layout_marginTop="10dp"
    android:background="@drawable/edit_border"
    android:fontFamily="@font/suitmedium"
    android:hint="할 일을 입력하세요."
    android:paddingStart="12dp"
    android:textSize="20sp"
    android:textStyle="normal"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/priorityTextview"
/>
```

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

Kotlin

Button을 눌렀을 때,
TODO를 저장시키는 것

SplashActivity(초기 화면)에서
MainActivity로
화면을 전환하는 것

Button을 눌렀을 때?
어떤 버튼인데?
라디오 버튼도 버튼 아닌가?

~~버튼이 무엇인지
정확하게 지정할 필요 있음~~

버튼을 눌렀을 때
코드를 작동시켜야 함

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin



MainActivity.kt

View.setOnClickListener { }

View 요소가 화면에서 클릭되었을 때,
{ } 안의 문장이 실행됩니다.

```
val addBtn = findViewById<Button>(R.id.mainToDoButton)

addBtn.setOnClickListener {
    var pr = 3
    if (radioBtn1.isChecked)
        pr = 0
    else if (radioBtn2.isChecked)
        pr = 1
    else if (radioBtn3.isChecked)
        pr = 2
    val contact = Contacts(addEditText.text.toString(), pr, false)
    runBlocking {
        db!!.contactsDao().insertAll(contact)
    }
    contactsList.clear()
    runBlocking {
        contactsList.addAll(db!!.contactsDao().getAll())
    }
    adapter.notifyDataSetChanged()
}
```

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

If = 만약에, ~라면

“만약 ~라면”이라는 의미로 작동합니다

```
if (조건식) {  
    수행 할 문장  
}
```

If = 만약에, ~라면 + else = 다른

위의 조건과 다른 조건이 성립할 때 수행하라는 의미로 작동합니다

```
else if (조건식) {  
    수행 할 문장  
}
```

조건식에 들어갈 문장	<, <=, >, >=, ==, != 관계(비교) 연산자	피연산자의 대소 관계를 나타낼 수 있는 연산자 값은 Boolean으로 나오며, true 혹은 false로 이야기함
	변수 is 자료형, 변수 !is 자료형	is) 변수의 자료형이 맞다면 True, 틀리다면 False !is) is의 반대 연산자로, 반대로 작동하는 연산자임
	범위를 나타내는 연산자	1 .. 100이라고 나타내게 된다면, 1부터 100까지의 범위

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

View.setOnClickListener { }

View 요소가 화면에서 클릭되었을 때,
{ } 안의 문장이 실행됩니다.

If = 만약에, ~라면

“만약 ~라면” 이라는 의미로 작동합니다

```
if (조건식) {  
    수행 할 문장  
}
```

If = 만약에, ~라면 + else = 다른

위의 조건과 다른 조건이 성립할 때 수행하라는 의미로 작동합니다

```
else if (조건식) {  
    수행 할 문장  
}
```

위를 이용하여 TODO 7~9를 해결 해봅시다!

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

Answer.
TODO 7

View.setOnClickListener { }

View 요소가 화면에서 클릭되었을 때,
{ } 안의 문장이 실행됩니다.

```
addBtn.setOnClickListener {  
    var pr = 3  
    if (radioBtn1.isChecked)  
        pr = 0  
    else if (radioBtn2.isChecked)  
        pr = 1  
    else if (radioBtn3.isChecked)  
        pr = 2  
    val contact = Contacts(addEditText.text.toString(), pr, false)  
    runBlocking {  
        db!!.contactsDao().insertAll(contact)  
    }  
    contactsList.clear()  
    runBlocking {  
        contactsList.addAll(db!!.contactsDao().getAll())  
    }  
    adapter.notifyDataSetChanged()  
}
```

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

Answer. TODO 8

If = 만약에, ~라면

“만약 ~라면”이라는 의미로 작동합니다

```
if (조건식) {  
    수행 할 문장  
}
```

If = 만약에, ~라면 + else = 다른

위의 조건과 다른 조건이 성립할 때 수행하라는 의미로 작동합니다

```
else if (조건식) {  
    수행 할 문장  
}
```

```
if (radioBtn1.isChecked)  
    pr = 0  
  
else if (radioBtn2.isChecked)  
    pr = 1  
  
else if (radioBtn3.isChecked)  
    pr = 2
```

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

Answer. TODO 9

```
val contact = Contacts(addEditText.text.toString(), pr, false)
```

위의 변수명은 자신이 findViewById 중 EditText를 저장시킨
요소를 가져오도록 합니다!

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

Kotlin

Button을 눌렀을 때,
TODO를 저장시키는 것

SplashActivity(초기 화면)에서
MainActivity로
화면을 전환하는 것

일정 시간이 지난 후,
화면을 전환 해주어야 함

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

val intent = Intent(this, Activity::class.java)
startActivity(intent)

Intent에 전환할 화면을 넣어주면 화면이 전환된다.



SplashActivity.kt

```
val intent = Intent(this, MainActivity::class.java)
intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION)
startActivity(intent)
```

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

```
val intent = Intent(this, Activity::class.java)  
startActivity(intent)
```

Intent에 전환할 화면을 넣어주면 화면이 전환된다.

위를 이용하여 TODO 10을 해결 해봅시다!

#01. 미니게임

#02. XML소개

#03. XML실습

#04. Kotlin

Answer.
TODO 10

```
val intent = Intent(this, Activity::class.java)  
startActivity(intent)
```

Intent에 전환할 화면을 넣어주면 화면이 전환된다.

```
val intent = Intent(this, MainActivity::class.java)  
intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION)  
startActivity(intent)
```

위의 PPT 내용을 다시 보고 싶다면
아래 링크를 접속해서 볼 수 있습니다

https://bit.ly/2022_SSF_DOCS

APP:PLE PI <나만의 TODO앱 만들기> 클래스

소프트웨어 나눔축제

수고하셨습니다 :)
궁금한 점이 있다면 아래로 문의주세요!



App:ple Pi 12기
최재혁

instagram
@jaehyeok3017



App:ple Pi 13기
문태서

instagram
@taeseo_06