

# 사랑의실천2

## [국문]

직무 역량 개발 계획서



학과	컴퓨터소프트웨어학부
학번	2024062806
이름	주하진

# 1. 직무 역량 개발 계획서

## 1-1. 나의 희망 직무와 역량

◎ 자신이 희망하는 직무 및 그와 연관 지을 수 있는 자신만의 강점과 취약점들을 적어보세요.

1) 나의 희망직무와 그 이유	<p>소프트웨어 개발자가 되고 싶었다. 초등학교 말미에 게임을 만들고 싶어서 구했던 파이썬 책을 보고 나서부터 그레왔다. 그때부터 나는 뭔가를 만들고 싶었다. 정교하게 설계된 코드 뭉치를 나는 만들고 싶었다.</p> <p>처음에는 내가 게임만 만들고 싶어할 줄 알았는데, 생각 외로 나는 촘촘하게 짜여진 소프트웨어 구조를 만들어 나가는 것이 재밌어졌다. 소프트웨어의 다양한 층위를 쌓아 올리는 것이 때로는 경이롭고 그것을 쌓아 올리는 내가 너무 재밌었다.</p>
2) 나의 강점	<p>나는 옛날부터 내가 궁금한것에 대해서 최대한 깊게 파고 들기를 잘하는 것 같다. 그 이면의 원리를 끝없이 파헤치는 걸 잘하는 것 같다.</p> <p>그리고 막히는 문제가 있으면 해결할 때까지 계속 찾아보는 걸 잘하는 것 같다. 말하자면 소위 ‘삽질’을 잘하는 것 같다. 어쩌면 끈기가 있다고도 말할 수 있을 것 같다. 나는 이 분야에서 ‘삽질’을 잘하는 것이 굉장히 큰 강점이고 장점이라고 생각한다. 왜냐하면 결국 무언가 쌓아 올릴 때 결국 시행착오는 필연적이기 때문에 그에 상응하는 ‘삽질’ 또한 필연적이기 때문이다.</p>
3) 나의 취약점 또는 보완할 점	<p>“언어”에서 되게 취약한 것 같다. 프로그래밍 언어가 아니라, “사람의 언어”. 애초에 그리고 어쩌면 공학적 재능을 어필하기 위해서는 언어적 재능이 필요하다. 어필은 본질적으로 다른 사람에게 보이게 하는 것이기 때문이다.</p> <p>그리고 이런 관점이 심화된 것은 요즘 소프트웨어가 만들어지는 과정에 대해서 생각해 보면서이다. 소프트웨어 제작 과정에서 “협업”은 매우 중요하다. 협업 관리 툴인 git이 왜 이렇게 유명해졌는가를 생각하면 매우 당연하다. 어쩌면 소프트웨어 제작에서 공학은 표현에 불과하고 언어야 말로 그 근본이라는 생각도 당연히 뒤따라왔다.</p> <p>나는 어쨋든 커뮤니케이션이 되게 심각한 취약점이라고 생각한다.</p>

## 1-2. 취업역량 개발 계획

◎ 단기 및 중, 장기적 측면에서 취업역량을 개발시킬 수 있는 구체적이고 실천적인 계획을 적어보세요.

<p><b>1) 직무역량 개발을 위한 구체적 실천 계획 - 단기</b></p>	<p>일단 최대한 많은 것을 경험해보려고 한다. 더 깊은 것도 물론 필요하지만 그래도 내 분야에 대한 넓은 지식이 필요하다고 느낀다. 그래서 요즘 컴파일러, 마이크로 프로세서나 AI, 서버, 리눅스, Android 개발, GUI 개발, 게임 개발 등 여러가지 지식을 습득하고 활용하려고 노력한다. 개인 프로젝트를 여러 개 만들어보고 게임 프로젝트를 하고 그렇게 실천하려고 한다.</p> <p>나중에는 다른 일이 많아질거고 최대한 저학년에 할 수 있는 공부를 하려고 한다.</p>
<p><b>2) 직무역량 개발을 위한 구체적 실천 계획 - 중, 장기</b></p>	<p>아직 대갈래만 정해진 것 같아서 정확히는 모르겠다. 이 업계는 내 생각에 무엇보다도 공부가 절실한 분야라고 생각한다. IT만큼 급변하는 분야가 어디 있을까? 그런만큼 이 곳에서 생존하려면 더 깊은 공부 및 연구가 필수적이라고 생각한다. 그런만큼 진학도 생각하고 있다. 아니면 실무에 관련된 학습을 하고서 프로그래머로서도 생각하고 있다. 이 두가지 생각에 공통된 점은 바로 공부가 끊임 없이 필수적이라는 문제다. 그만큼 중장기적으로 무엇에 대한 공부는 달라질 수 있어도 끊임 없이 달려야 한다.</p> <p>그리고 대부분의 지식 수급 언어는 영어인만큼 영어에 대한 공부도 끊임 없이 필요하다. 논문 등을 읽으려면 말이다.</p>

## 2. 한양역량진단검사

◎ 한양역량진단검사 3 가지를 실시한 후, 검사결과를 정확하게 첨부하세요.

### 2-1. My Career Status Test(나의 희망진로현황)

\* HY-IN ▶ 취업 ▶ HY-CDP ▶ 진단실시 ▶ 나의 희망진로현황 ▶ 검사실시

2차 2학년 / 2025-11-17 실시

진로현황조사결과

결과보기

취업 40%

대학원 40%

창업 20%

### 2-2. Hanyang Leadership Competency(한양리더십역량진단)

\* HY-IN ▶ 취업 ▶ HY-CDP ▶ 진단실시 ▶ 한양리더십역량진단 ▶ 검사실시



# 한양리더십(HELP) 역량 진단평가

HELP Competency Test



· 학교명 **한양대학교**  
 · 학과명 **컴퓨터소프트웨어학부**  
 · 학번 **2024062806**  
 · 성명/성별 **주하진/남자**

## 2학년

**— 주하진님**  
**— 한양대학교 평균**



척도	점수	등급	해석
창의력	3.67	중	창의력이 다른 학생들과 비슷한 수준입니다. 기존의 관행이나 방식을 뛰어넘을 수 있는 아이디어 발상을 꾸준히 실천해 보세요.
문제 해결력	4.33	상	문제해결능력이 다른 학생들에 비해 높은 수준입니다. 문제 발생시 문제의 본질을 잘 파악하고, 이를 구조화, 단순화하여 합리적으로 해결할 수 있습니다.
학습능력	4.67	상	학습능력이 다른 학생들이 비해 높은 수준입니다. 자신의 능력과 지식을 점기적으로 점검하여 지속적인 학습을 통해 지식과 경험을 업그레이드하고 있습니다.
글로벌 트렌드	4.33	상	글로벌 트렌드에 대한 지식이 다른 학생들에 비해 많은 편입니다. 세계적으로 이슈가 되는 트렌드를 통해 중장기적인 관점에서 변화의 방향을 읽을 수 있으며, 이런 통찰력을 통해 나의 미래를 준비하기 위한 다양한 아이디어를 찾을 수 있을 것입니다.
최고지향	4	하	최고가 되고자 하는 강한 의지를 가질 필요가 있으며, 구체적인 목표와 계획을 수립하고 실행할 수 있는 역량이 필요합니다.
글로벌 마인드	3	하	글로벌 마인드 보유수준이 다른 학생들에 비해 다소 낮은 편입니다. 문화적인 충돌이나 갈등상황에서 유연하게 잘 대처할 수 있도록 나와 다른 배경이나 문화를 가진 사람들과 보다 더 적극적인 대화가 필요합니다.
커뮤니케이션	2.33	하	커뮤니케이션 역량 보유 수준이 다른 학생들에 비해 다소 낮은 편입니다. 상대방의 말에 주관적인 판단 없이 객관적으로 경청하는 태도가 필요하며, 다른 사람에게 자신의 의사를 사실 없이 전달하게 전달할 수 있는 방법을 학습해 보세요.

### 2-3. S-Hanyang Core Competency(S-한양핵심역량)

\* HY-IN ▶ 학사행정 ▶ S-한양핵심역량 ▶ 검사실시 및 결과 ▶ 검사실시

인재상	역량	T점수			평가	강점/약점
교양인	자기인식	본인		46.1	주하진 학생의 자기인식 역량 검사 결과는 보통 수준으로 나타났습니다. 이는 한양대학교 전체 재학생 중 하위 30~50% 사이에 해당됩니다. 자기인식 역량은 인문학, 사회과학, 자연과학 등 폭넓은 관점에서 스스로 돌아봄으로써 자신의 특성과 장단점을 이해하고 지속적으로 성장해 나가려는 역량입니다.	
		학과평균		54.0		
		학교평균		54.6		
	자율책임	본인		49.7	주하진 학생의 자율책임 역량 검사 결과는 보통 수준으로 나타났습니다. 이는 한양대학교 전체 재학생 중 하위 30~50% 사이에 해당됩니다. 자율책임 역량은 자신이 나아가야 할 방향을 스스로 설정하고 자주적으로 실행하며 행동의 결과를 받아들이는 역량입니다.	
		학과평균		52.4		
		학교평균		53.0		
전문인	지식	본인		69.5	주하진 학생의 지식 역량 검사 결과는 매우 탁월한 수준으로 나타났습니다. 이는 한양대학교 전체 재학생 중 상위 10% 이내에 해당됩니다. 지식 역량은 학문적 이론과 방법론을 폭넓게 탐구하고 다양한 정보를 다각적으로 수집 및 활용하는 역량입니다.	강점
		학과평균		52.7		
		학교평균		52.0		
	비판적사고	본인		62.9	주하진 학생의 비판적사고 역량 검사 결과는 매우 탁월한 수준으로 나타났습니다. 이는 한양대학교 전체 재학생 중 상위 10% 이내에 해당됩니다. 비판적사고 역량은 어떤 정보나 주장, 신념을 합리적이고 반성적(反省的)으로 분석, 평가, 추론하는 역량입니다.	강점
		학과평균		56.5		
		학교평균		54.8		
실용인	창의	본인		59.1	주하진 학생의 창의 역량 검사 결과는 탁월한 수준으로 나타났습니다. 이는 한양대학교 전체 재학생 중 상위 10~30% 사이에 해당됩니다. 창의 역량은 다양한 관점에서 문제를 발견하여 새롭고 유용한 아이디어를 통해 문제를 해결하고자 하는 역량입니다.	강점
		학과평균		53.7		
		학교평균		52.9		
	도전	본인		57.9	주하진 학생의 도전 역량 검사 결과는 탁월한 수준으로 나타났습니다. 이는 한양대학교 전체 재학생 중 상위 10~30% 사이에 해당됩니다. 도전 역량은 높은 수준의 목표를 설정하여 실패를 두려워하지 않으며 어떠한 어려움도 극복하려는 자세를 갖추고, 익숙지 않거나 새로운 일을 시도하는 역량입니다.	
		학과평균		52.2		
		학교평균		52.8		
세계인	소통	본인		33.9	주하진 학생의 소통 역량 검사 결과는 매우 미흡한 수준으로 나타났습니다. 이는 한양대학교 전체 재학생 중 하위 10% 이내에 해당됩니다. 소통 역량은 언어학, 비언어학 방식으로 타인의 생각, 감정, 다른 문화를 이해하고 자신의 생각과 감정을 표현하는 역량입니다.	약점
		학과평균		51.0		
		학교평균		52.9		
	협업	본인		50.9	주하진 학생의 협업 역량 검사 결과는 우수한 수준으로 나타났습니다. 이는 한양대학교 전체 재학생 중 상위 30~50% 사이에 해당됩니다. 협업 역량은 다른 문화권의 사람들을 포함하여 다양한 사람들과 신뢰관계를 구축하고 역할을 나누어 효과적으로 공동의 목표 달성을 기여하는 역량입니다.	
		학과평균		51.6		
		학교평균		52.8		
봉사인	공감	본인		44.8	주하진 학생의 공감 역량 검사 결과는 보통 수준으로 나타났습니다. 이는 한양대학교 전체 재학생 중 하위 30~50% 사이에 해당됩니다. 공감 역량은 타인의 생각, 감정, 가치관에 친近한 관심을 가지고 깊이 있게 이해하며 경서적으로 교감하는 역량입니다.	약점
		학과평균		51.6		
		학교평균		52.9		
	나눔	본인		51.8	주하진 학생의 나눔 역량 검사 결과는 우수한 수준으로 나타났습니다. 이는 한양대학교 전체 재학생 중 상위 30~50% 사이에 해당됩니다. 나눔 역량은 공동체적 유대감을 바탕으로 자신이 가진 것을 사회적 약자를 포함한 사회구성원 전체와 함께 나누며 사랑을 실천하는 역량입니다.	
		학과평균		50.5		
		학교평균		52.5		

### 3. MOOC Self-Directed Learning

## MOOC Self-Directed Learning (1)

### ：선정 배경 및 학습 계획

사이트	http://www.kocw.net/home/search/kemView.do?kemId=1488273		
분야	공학 >컴퓨터 · 통신 >컴퓨터공학	강사명	오병우
강의명	Jetpack Compose를 사용한 안드로이드 프로그래밍	강의시간	40~60분 영상 22개

#### 강의 선정 배경

평소부터 Android 개발을 하고 싶었다. 단순히 어떤 일 또는 학업적인 문제가 아니라 개인적인 문제로. 이유는 내가 대부분의 시간에 들고 있는 폰이라는 것을 더 잘 사용하기 위해서이다. 이런 자신만의 애플리케이션을 만들어보는 것에 관심이 있었기 때문이다. 그런데, 자꾸 Android 개발을 생각만하고 제대로 해본적이 없다. Android 개발에서 필요한 것이 너무 복잡 했기 때문이다. 프레임워크의 복잡도는 진짜 생각이상으로 너무 컸다. 그래서 이번기회에 Android 개발을 시작해보고자 선정 했다.

사실 비단 개인적인 목적이 아니라고 해도 대부분 프레임워크는 많은 점을 공유하고 있기 때문에 이것, Jetpack Compose를 배워두고 나면 두고두고 쓸 일이 있을 것 같다. 또 적어도 10+년동안은 Android가 없어질 일은 없지 않을까?

#### 강좌 자기주도학습 계획(안)

1주차: Kotlin 프로그래밍이 처음이 아니기 때문에, 2차시부터 진행, 대충 UI Element를 배치해보는 연습을 할 듯, Layout하고

2주차: State와 Composable, 그리고 중요하다고 생각하는 Navigation에 대해서 배울 듯

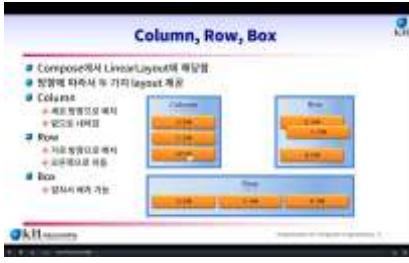
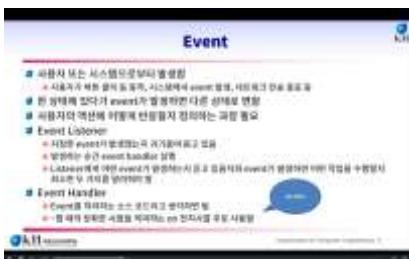
3주차: ViewModel과 데이터 처리를 공부할 듯

4주차: Intent에 대해서 배워볼 것 같다. Supabase와 Pocketbase를 배우기보단 (이런 서비스를 알려고하기 보단 본질을더 알고 싶다, 지금 당장 서비스를 만들 것도 아니고) SQLite를 위주로 볼 듯 그리고 Retrofit을 배워볼거 같다. 서버와 통신하는 API를 알기 위해서.

# MOOC Self-Directed Learning (2)

## ： 요약 및 학습노트

◎ 자기주도학습을 통해 강의 내용 요약과 학습노트(나에게 중요했던 점, 시사점)를 적어보세요.

강의 화면	강의 내용 요약 및 학습노트
	<p>Android Studio에서 프로젝트를 만드는 방법, 프로젝트 이름과 패키지 이름(영어 소문자로 시작), API 버전(Android Q, S, R 등)에 대한 내용.</p> <p>기본 제공 Greeting을 수정함. NamedParameter, modifier를 넘기는 방법, 폰트를 바꾸는 방법 chaining이 됨. (modifier.padding(30.dp).background(Color.blue))</p> <p>Modifier는 순서가 중요함.</p> <p>Image도 한번 넣어봄. painterResource를 이용해서 이미지 애셋을 넣기. 접근성이 중요해서 description도 넣어야함. 그리고 Image에 대한 몇가지 property. 또 modifier clickable도 넣음.</p> <p>Linear Layout: Box, Row, Cloumn을 써봄. Programmer가 처리하기 쉬움.</p>
 	<p>Modifier: Composable의 모양 및 레이아웃 사용자 입력 처리 등등을 하는 요소. Chaining이 가능하며 순서에 따라 달라짐. 기능 및 요소로는 size, padding, offset, background clickable 등등이 있다. Modifier는 Immutable하다.</p> <p>Layout(개인적으로 중요): Composable이 화면에 어떻게 배치되는지 정의하는 Composable로 Row, Column, Box (모두 LinearLayout) 등이 있다.</p> <p>Event처리에 대해서 배움 `setOnClickListener`와 같은 이벤트 리스너 구조를 봄. Android에서 UI는 XML을 자주 씀.</p> <p>대부분의 UI 프레임워크는 비슷한 구조를 띄기 때문에 다른 곳에서도 쓸 수 있음.</p> <p>이벤트 처리는 상태 전이로 표현이 됨.</p> <p>Android Jetpack Compose에서는 State를 씀. (모든 변수를 처리하는 건 비효율적이기 때문)</p> <p>State를 정의할 때 remember를 씀 이것은 mutable함.</p> <p>Composable은 기본 요소로 다른 프레임워크에서는 Widget, Component 등으로 불림. Composable Function은 일반적으로 PascalCase를 사용함. 파라미터를 받을 수 있다 그리고 trailing lambda를 사용할 수 있도록 마지막 param을 함수로 지정할 수 있음. Return value는 Unit(Void). Composable 내에 또다른 Composable을 실행시킬 수 있음.</p> <p>Recomposition: State와 Event는 Composable과 매우 밀접한 관계를 갖고 있음. State가 바뀌면 Composable을 다시 실행함. 변경이 필요한 부분만 선택적으로 다시</p>

	<p>실행.</p> <p>앱 개발 중에는 화면을 변경해야 할 수 있음. Compose에서는 Navigation으로 화면을 전환함. Activity를 추가할 수 있지만, Compose에서는 Activity를 추가하는 방식을 권장하진 않음. 일반적으로 한 개의 Activity를 이용하고 여러 화면을 이용해야되면 Navigation을 이용함. (SPA)</p> <p>이 화면은 본질적으로 Composable. 이 Composable 간에 넘어갈 수 있는 방법을 제공하는 것이 Navigation.</p> <p>Navigation은 NavController, NavGraph, NavHost로 구성됨</p> <p>NavController는 일종의 관리자로 가장 윗레벨의 상태를 가지고 있음.</p> <p>NavGraph는 navigation을 관리하기 위해 graph를 사용한다. 화면들이 어떻게 연결돼있는지 알기 위한 데이터 구조.</p> <p>NavHost는 Composable로 현재 화면임.</p>
	<p>Intent는 실행하려는 의지를 담은 일종의 메시지 객체임. 적절한 Activity를 실행함.</p> <p>XML 방식에서는 필수적이었음 왜냐하면 Activity가 여러 개였기 때문.</p> <p>Intent에는 Implicit와 Explicit가 있음.</p> <p>Explicit Intent: 의도에 맞는 Activity Class를 지정하여 실행시키는 방식으로, Activity Class 이름을 알고 있어야 함. XML Views에서 화면 전환을 위해 주로 사용함.</p> <p>Implicit: 의도만 주고 알아서 운영체제한테 알아서 해달라고 함. Action을 지정하여 Intent 객체를 생성하고 실행해달라고 하면 운영체제는 알아서 찾음.</p> <p>IntentFilter의 Action에 맞는 Activity를 사용. 이는 Android Manifest.xml에서 정의됨.</p> <pre>val intent = Intent(Intent.ACTION_VIEW, uri)</pre> <p>DB를 사용하는 것이 편하다. SQLite를 사용. Room은 SQLite의 라이브러리 이름. 일반적으로 DB는 서버에 있지만, 내부에서도 쓸 수 있다. 그게 SQLite.</p> <p>Entity는 개체, 실제 개체를 의미할 수도 하지만 DB에서는 추상화하여 몇가지 프로퍼티만 정의할 수 있다. 일종의 Table과 같다.</p> <p>Database는 영속적인 (persistent) 데이터를 모아놓고 관리하는 단위. 메모리와 비교됨. DB는 테이블을 관리함. DAO는 DB에 접근하기 위해 사용하는 객체로 Query를 DB에 접근하여 처리(CRUD)</p> <p>Flow는 생산자로부터 데이터가 흘러갈 수 있는 비동기 stream</p> <p>생산자 소비자 구조를 사용함. 위 Room Library를 이용해 화면을 디스플레이할 때 생산자는 Room Library 소비자는 Composable 등</p>

Retrofit은 백엔드에 접근하기 위한 라이브러리로 HTTP API를 Interface로 정의하면  
자동으로 구현해주는 라이브러리  
Json과 Gson Retrofit에서 Gson을 이용

## MOOC Self-Directed Learning (3)

### : AI x QBL(Question-Based Learning) 활용

◎ 질문(QBL: Question-based Learning)과 생성형 AI들을 활용하여 향후 탐구하고 싶은 분야를 구체화하거나 사고를 확장해봅시다.



작성 방법		
① 나의 질문	② 생성형 AI의 답변 요약	③ 나의 피드백(재질문)
학습을 통해 생겨난 호기심, 해결할 문제, 탐구 분야 등 구체적 질문	▶ 질문에 대해 받은 답변 요약	▶ AI 답변에 대해 나의 의견, 생각과 함께 추가 질문을 하여 피드백하기
나의 질문	Jetpack Compose에서 가장 핵심은 Composable인 것으로 보인다. 내가 조금 더 찾아 본 결과 UI를 선언적으로 정의하는 패러다임을 충실히 구현하는데 매우 중요한 역할을 한다. Composable은 State를 이용해서 업데이트 해야 할 것과 그려지 않을 것을 관리하여 효율적으로 렌더링을 진행한다. 그러면 Composable의 구현은 어떻게 이루어지는가? 이를 구현하는 Jetpack Compose의 구조를 상세하게 설명해주거라.	

<b>생성형 AI의 답변 요약</b>	<p>Composable 함수의 구현과 런타임 구조는 <b>Composer</b>와 <b>Slot Table</b>을 중심으로 작동합니다.</p> <ol style="list-style-type: none"> <li><b>@Composable 어노테이션:</b> 일반 Kotlin 함수를 Composable 함수로 변환하도록 컴파일러에 지시합니다.</li> <li><b>컴파일러 변환:</b> 모든 Composable 함수에 내부적으로 <b>Composer</b> 객체와 메타데이터를 추가합니다.</li> <li><b>Composition (구성):</b> Composable 트리가 <b>최초 실행될 때</b>, Composer는 모든 호출과 생성된 UI 구조를 <b>Slot Table</b>에 기록합니다.</li> <li><b>Slot Table:</b> UI 요소의 계층 구조, 위치, 그리고 상태 데이터를 저장하는 핵심 메모리 구조입니다.</li> <li><b>State 변경:</b> <code>remember { mutableStateOf(...)</code> 등으로 관리되는 <b>State</b> 값이 변경되면 Recomposition이 트리거됩니다.</li> <li><b>Recomposition (재구성):</b> 변경된 State를 사용하는 Composable 함수만 <b>다시 실행하는 과정입니다.</b></li> <li><b>Skipping:</b> Compose 런타임은 Slot Table의 기존 데이터와 비교하여 <b>변경되지 않은 Composable</b>의 실행을 건너뜁니다.</li> <li><b>안정성 (Stability):</b> 매개변수의 타입이 안정적일 경우에만 Skipping이 가능하여 효율성이 극대화됩니다.</li> <li><b>업데이트 반영:</b> 재실행된 결과 중 <b>실제 변경된 부분만</b> 화면의 네이티브 UI 트리에 반영됩니다.</li> <li><b>선언적 구현:</b> 이 구조를 통해 개발자가 UI의 <b>상태만</b> 선언하면, Compose가 <b>효율적인 업데이트 과정을 책임지게 됩니다.</b></li> </ol>
<b>AI답변에 대한 나의 피드백 (재질문)</b>	<p>Composable의 상태데이터의 변경은 어떻게 감지 하는가? Remember는 어떻게 구현하는가?</p> <p>Recomposition은 어떻게 진행될까 process 또는 실제 내부적 동작방식에 대해서 궁금함.</p> <p>안정성은 어떻게 구분할까?</p> <p>Slot Table은 어떻게 구현되는가?</p> <p>선언적 구현을 다른 언어 또는 프레임워크에서 어떻게 구현할까? 예를 들어서 파이썬에서는 어떻게 구현할까? 아니면 내가 요즘 관심 있는 Rust에서는 구현할 수 있을까?</p>

# MOOC Self-Directed Learning (4)

## ： 학업/직무 시사점 및 향후 계획

◎ MOOC 자기주도학습을 통해 느낀 점, 나의 학업/직무 관련 시사점, 궁금한 점, 향후 계획 등을 적어보세요.

구분	각 항목 작성
MOOC 자기주도학습을 통해 느낀 점	<p>Jetpack Compose/Kotlin을 이용해서 대강의 Android Application 개발을 충실히하게 할 수 있도록 하였다. 나는 이를 통해서 Application을 개발할 수 있는 지식을 어느정도는 습득한 것 같다.</p> <p>이와 별개로 MOOC이라는 시스템이 되게 괜찮은 것 같다. 내가 생각지도 못했던 강의를 들을 수 있는 시스템이 매우 좋은 것 같다.</p> <p>그리고 이렇게 몇 주 잡아서 강의를 듣고 정리하고 실습해보는 경험이 의외로 흔치 않아서 (보통은 강의나 수업만 들었으니까), 색다르기도 하고 재밌다.</p>
나의 학업/직무 관련 시사점	<p>학업적 직무적 시사점은 솔직히 말하면 엄청 크다고 생각하지는 않다. 내가 UI 관련 개발을 빡세게 할까라는 원론적인 의문도 남아있다. 그러나 UI 개발은 개인적으로 프로그래밍과 떼어 놓을 수 없는 것이라고 생각하기도 한다. 그래서 도움이 될거라고 믿는다.</p> <p>UI 개발 애플리케이션 개발이라는 제한된 영역 말고, HTTP 호출, MVVM, 그리고 이벤트 리스너 같은 흔히 쓰이는 패턴 학습이라는 측면에서 봤을 때 매우 중요하다고 생각한다. 그리고 그 이면에 쓰이는 패러다임에 대한 학습도 매우 시사하는 바가 크다.</p>
향후 계획	<p>좀 더 UI의 핵심적인 경험, 심화적인 응용을 해보고 싶기도 하다. 지금 배운건 진짜 기초적이라고 느끼는 바가 있기 때문이다.</p> <p>만약 향후에 시간이 된다면 선언적 UI 프레임워크에 대해 알아보고 내가 직접 그런 라이브러리를 만들어보고 싶기도 하다.</p> <p>사실 UI를 구현하는데 이런 프레임워크 뿐만이 아니라 게임엔진을 사용할 수도 있다. 그래서 게임엔진을 이용한 UI개발, 또는 Android 내에서 Vulkan, OpenGL 등을 사용하는 경험도 이왕이면 해봤으면 좋겠다.</p>