

# 5주차 - 17차 스터디 중간 세미나

주제: 프로그래머로서 지속 가능한 삶

일시: 2019년 10월 2일 수요일 20:00 ~ 22:00

장소:

- 우아한형제들 작은집 7층 (<http://naver.me/5hZ1c0R5>)
- 우아한형제들 루터회관 14층 (<http://naver.me/IFebBJNC>)

시간표:

시간	Track 1	Track 2	Hands-On Lab
	작은집 7층 교육장	루터회관 14층 트랙방	루터회관 14층 교육장
20:00 ~ 20:45	Multi Thread Server vs Single Thread Server 박재성(자바지기)	게임 재화 및 재화 로그 시스템 개발기 김문수	Tensorflow 시작하기 1개월 천정대
20:45 ~ 21:00	휴식		
21:00 ~ 21:45	MongoDB for Big Data Processing 김민호	이직 트리거 : 나는 왜 그 회사를 떠났나 이태훈	코틀린(Kotlin) DSL로 깔끔한 API 만들기 박재성(Jason)

기타:

- 끝나고 간단한 뒷풀이 예정

세션 소개:

## 1. Multi Thread Server vs Single Thread Server - 박재성(자바지기)

- 최근 reactive 개념이 활성화되면서 single thread server 기반 개발이 활성화되고 있다. single thread server 기반이 되는 non-blocking, async 개념을 이해하기 위해 sync vs async, blocking vs non-blocking 개념에 대해 이해하고, single thread server 동작 원리를 이해해 본다.
- [#sync](#) [#async](#) [#blocking](#) [#non-blocking](#)

## 2. MongoDB for Big Data Processing - 김민호

- 대용량 데이터를 저장하고 검색하기 위한 다양한 데이터베이스가 있습니다. 그 중에 몽고디비에 관해 살펴보고자 합니다. 대용량 데이터를 저장하고 분산 처리하기 위한 사딩과 고가용성을 위한 레플리카셋 그리고 다양한 몽고디비의 특징을 알아봅니다. 또한 이러한 대용량 데이터를 저장하기 위해 데이터 모델을 선정하고 대용량 데이터 분석 처리를 위한 애플리케이션 개발 경험과 몽고디비 샤드 클러스터를 직접 운영해본 경험도 함께 공유할 예정입니다.
- [#문서데이터베이스](#) [#몽고디비](#) [#샤딩](#) [#레플리카셋](#) [#대용량데이터](#) [#분산처리](#) [#NoSQL](#)

## 3. 게임 재화 및 재화 로그 시스템 개발기 - 김문수

- 모바일 게임을 개발하면서 난관 중 하나였던 게임 내의 재화와 그것의 로그를 어떻게 설계하고 개발했는지 공유해 봅니다. 작은 영감을 얻어 갈 수도, 혹은 제게 다른 방향을 제시해줄 수도 있을 수 있습니다. 한 번 이야기해 보아요.
- [#SRP설계의중요성](#) [#AOP활용기](#) [#ThreadLocal활용기](#)

## 4. 이직 트리거 : 나는 왜 그 회사를 떠났나 - 이태훈

- 이직을 4번 하면서 왜 결심하게 되었는지 그리고 매번 마음에 남는 말 한마디가 있었습니다. 각 회사에서 어떻게 성장하거나 정체되었고 무엇을 얻고자 이직하였는지 이야기 합니다.
- [#이직](#) [#성장](#) [#살아온이야기](#)

## 5. Tensorflow 시작하기 1개월 - 천정대

- 왜 Tensorflow 인가?  
- Tensorflow 공부 방법 삼질기  
- 머신러닝, 딥러닝, Tensorflow 소개  
- Tensorflow 상위 툴 Keras 소개  
- 웹 브라우저 또는 노트북에서 Tensorflow 실행해 보기  
- ANN, DNN, CNN, RNN, AE, GAN, UNET 소개  
- 간단 예제 따라해 보기
- [#핸즈온랩](#) [#딥러닝](#) [#Tensorflow](#) [#Keras](#)

## 6. 코틀린(Kotlin) DSL로 깔끔한 API 만들기 - 박재성(Jason)

- 명확하고 간결한 코드 작성을 위해 코틀린은 확장 함수, 중위 호출, 연산자 오버로딩 등 다양한 기능을 제공합니다. 이번 시간에는 도메인 특화 언어(DSL, Domain-specific language)를 사용해 표현력이 좋고 코틀린다운 API를 만들어 보겠습니다.
- [#핸즈온랩](#) [#코틀린](#) [#깔끔한](#) [#API](#) [#DSL](#) [#수신객체지정람다](#) [#invoke](#) [#빌드스크립트](#) [#GradleKotlinDSL](#) [#데이터베이스](#) [#HTML](#) [#테스트](#)