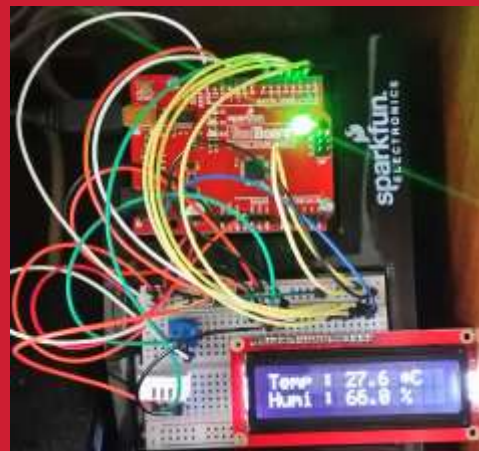




# Arduino-IoT

[wk12]

## Arduino + Node Data storing I



Visualization of Signals using Arduino,  
Node.js & storing signals in MongoDB  
& mining data using Python



Drone-IoT-Comsi, INJE University

2<sup>nd</sup> semester, 2020

Email : chaos21c@gmail.com



# My ID

## 1분반-목요일 (2학년)

- AA1-01: 강서현
- AA1-02: 강태민
- AA1-03: 김세은
- AA1-04: 여수민
- AA1-05: 정영훈
- AA1-06: 차혁준
- AA1-07: 하태현
- AA1-08: 김경욱
- AA1-09: 김민욱
- AA1-10: 김민성
- AA1-11: 김민준
- AA1-12: 김인수
- AA1-13: 김현식
- AA1-14: 장성운
- AA1-15: 전승진
- AA1-16: 정희철
- AA1-17: 조동현
- AA1-18: 전동빈
- AA1-19: 신종원

## 2분반-수요일 (3학년)

- AA2-01: 강민수
- AA2-02: 구병준
- AA2-03: 김종민
- AA2-04: 박성철
- AA2-05: 이승현
- AA2-06: 이창호
- AA2-07: 손성빈
- AA2-08: 안예찬
- AA2-09: 유종인
- AA2-10: 이석민
- AA2-11: 이정문
- AA2-12: 이주원
- AA2-13: 정재영
- AA2-14: 하태성
- AA2-15: 김경미
- AA2-16: 김규년
- AA2-17: 김유빈
- AA2-18: 송다은
- AA2-19: 정주은
- AA2-20: 권준표



# [Review]

## ◆ [wk11]

- RT Data Visualization with node.js
- Multiple data and Usage of gauge.js
- Complete your real-time WEB charts
- Upload folder: aax-nn-rpt09
- Use repo “aax-nn” in github

# wk11 : Practice : aax-nn-rpt09

## ◆ [Target of this week]

- Complete your works
- Save your outcomes and upload outputs in github

제출폴더명 : **aax-nn-rpt09**

- 제출할 파일들

- ① **AAnn\_cds\_dht22\_data.png**
- ② **AAnn\_cds\_dht22.html**
- ③ **AAnn\_cds\_dht22.png**
- ④ **All \*.ino**
- ⑤ **All \*.js**
- ⑥ **All \*.html**



# IOT: HSC

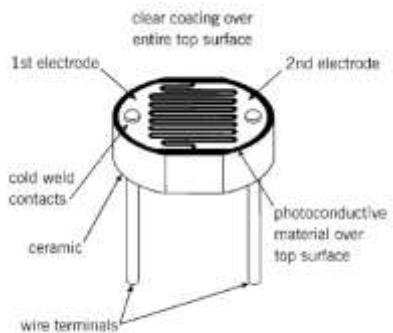
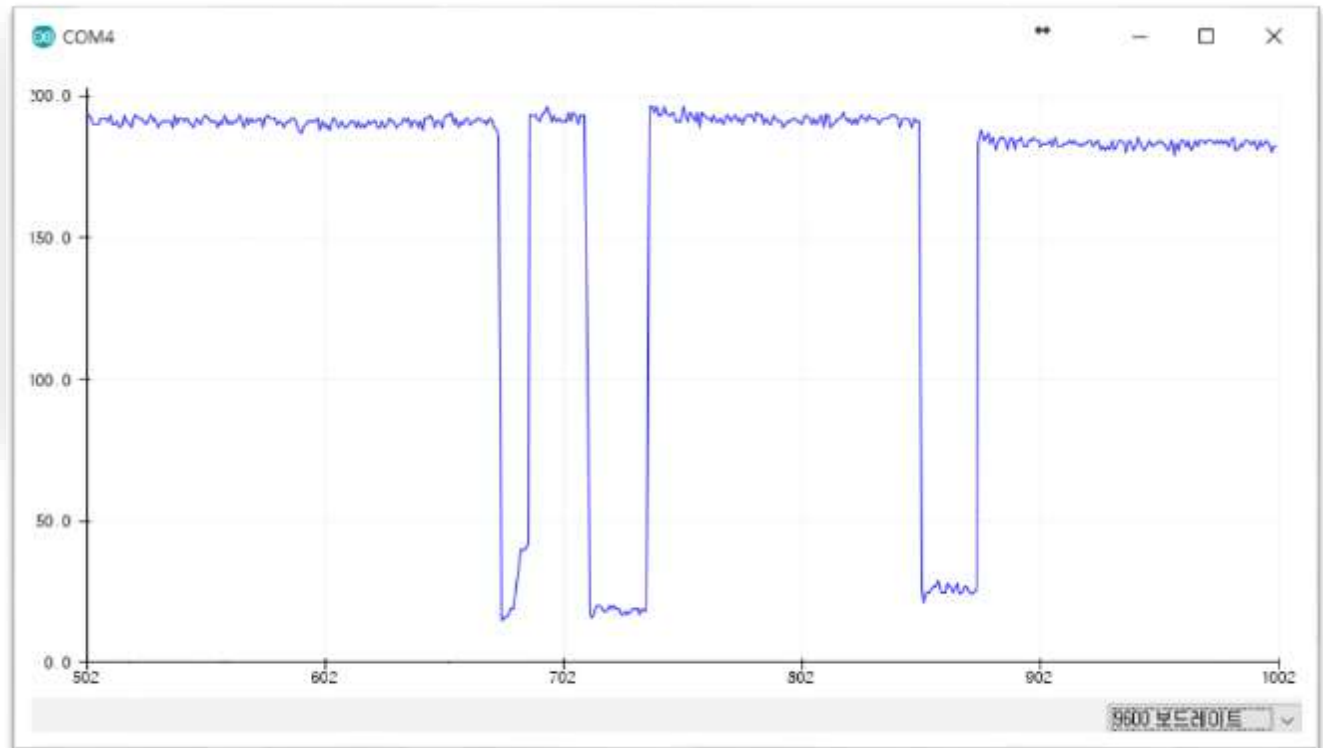
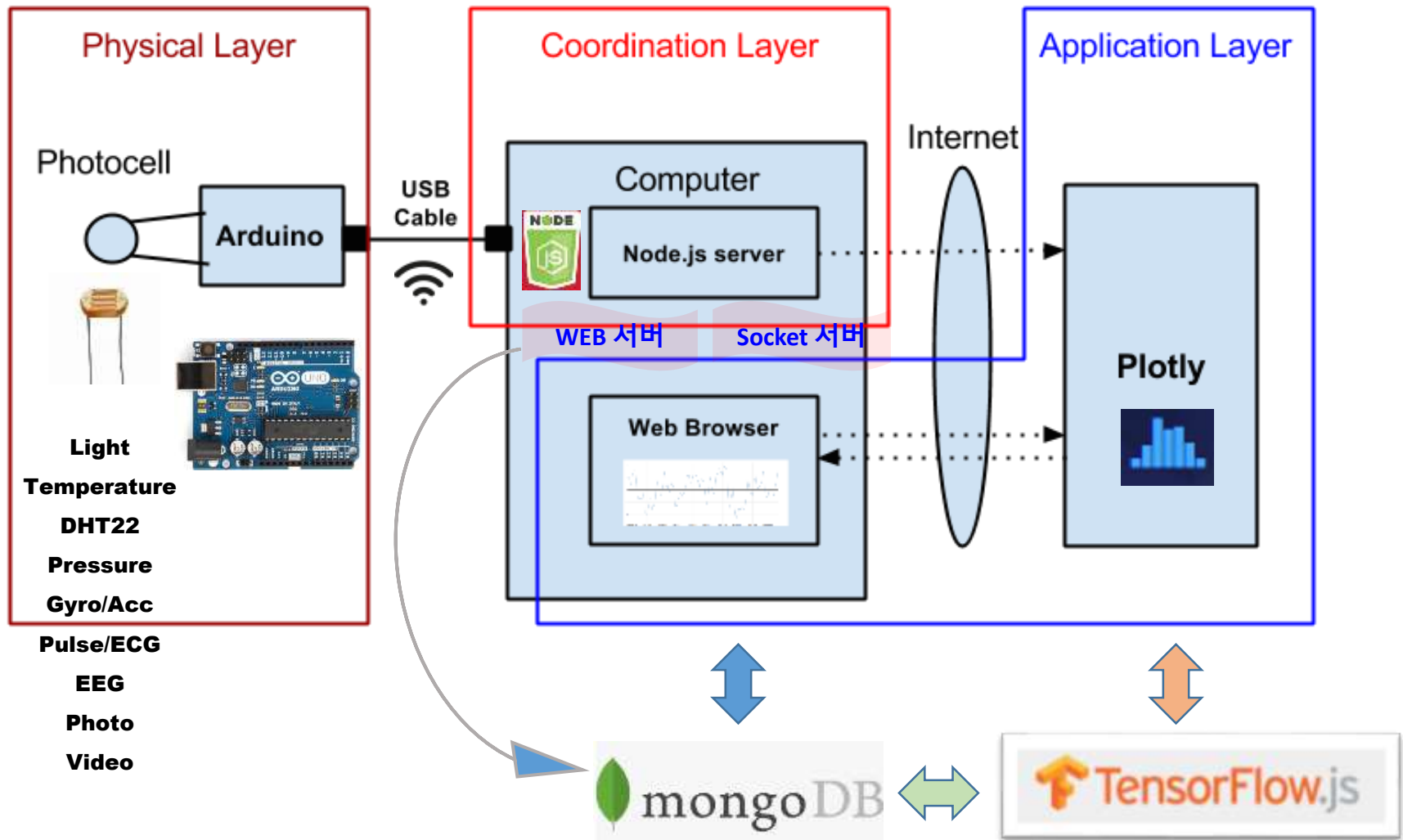


Figure 3  
Typical Construction of a Plastic Coated Photocell



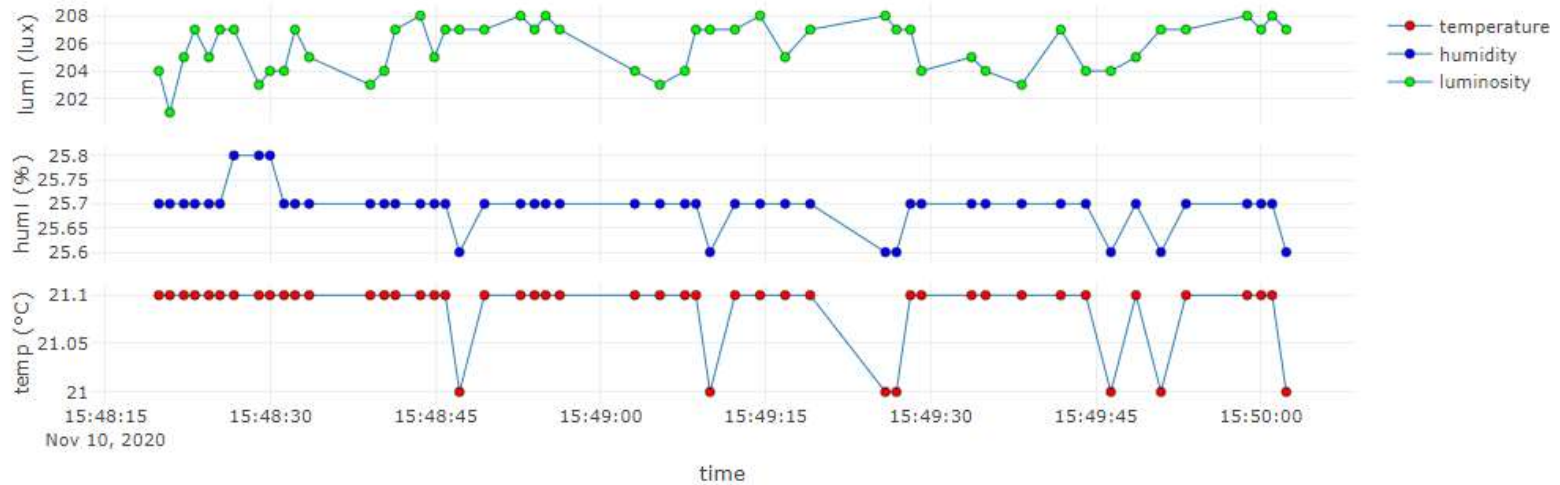
# Layout [H S C]



# Real-time Weather Station from sensors



on Time: 2020-11-10 15:50:02.300





## A5. Introduction to IoT service

**System (Arduino, sDevice, ...)**



**Data (signal, image, sns, ...)**



**Visualization & monitoring**

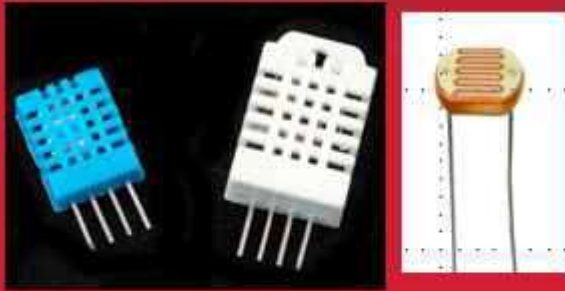


**Data storing & mining**



**Service**





[Goal]

Arduino + Node.js

+ plotly.js

+ MongoDB

→ Data storaging

& visualization



# A5.9 MongoDB




MongoDB for GIANT id x

← → ↻ 🏠 | 안전함 | https://www.mongodb.com

DOCS LEARN WHAT'S MONGODB? LOGIN

📞 🔍 [Free Sandbox](#) [Download](#)

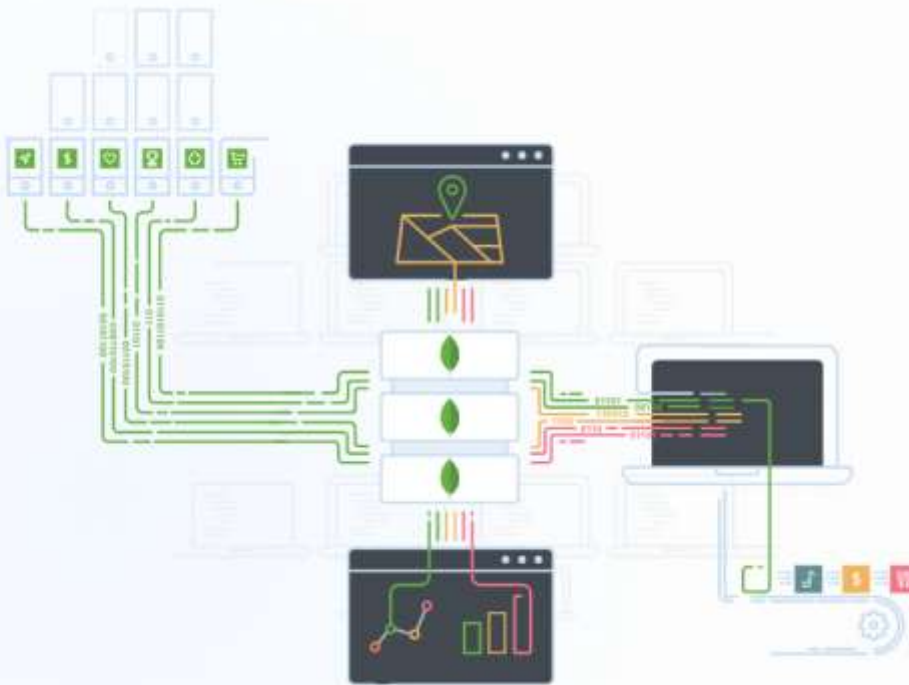
**mongoDB.** | FOR GIANT IDEAS SOLUTIONS CLOUD CUSTOMERS RESOURCES ABOUT US

 **mongoDB®**

**Move at the Speed of Your Data**

Go faster with MongoDB 3.6

[Learn more](#)





# A5.9 MongoDB



**MongoDB**는 **C++**로 작성된 오픈소스 문서지향(**Document-Oriented**) 적 **Cross-platform** 데이터베이스이며, 뛰어난 확장성과 성능을 자랑합니다. 또한, 현존하는 **NoSQL** 데이터베이스 중 인지도 1위를 유지하고 있습니다.

## NoSQL?

흔히 **NoSQL**이라고 해서 아, **SQL**이 없는 데이터베이스구나! 라고 생각 할 수도 있겠지만, 진짜 의미는 **Not Only SQL** 입니다. 기존의 **RDBMS**의 한계를 극복하기 위해 만들어진 새로운 형태의 데이터저장소 입니다. **관계형 DB**가 아니므로, **RDMS**처럼 고정된 스키마 및 **JOIN**이 존재하지 않습니다.

## Document?

**Document Oriented** 데이터베이스라는데.. 여기서 말하는 **Document**가 뭘까요? 문서? 이게 그냥 '문서'로 번역해버리면 조금은 애매합니다. 문서라고 하면 보통 워드/엑셀에 사용되는 그런 문서가 떠오르는데요, 그것과는 다릅니다. **Document**는 **RDMS**의 **record**와 비슷한 개념인데요, 이의 데이터 구조는 한개이상의 **key-value pair**으로 이루어져 있습니다. **MongoDB** 샘플 **Document**를 확인해 볼까요?

```
{ "_id": ObjectId("5099803df3f4948bd2f98391"),
  "username": "velopert",
  "name": { first: "M.J.", last: "Kim" } }
```



# A5.9 MongoDB



여기서 **\_id, username, name** 은 **key** 이고 그 오른쪽에 있는 값들은 **value** 입니다.

**\_id** 는 12bytes의 hexadecimal 값으로서, 각 document의 유일함(uniqueness)을 제공합니다.

이 값의 첫 4bytes 는 현재 timestamp, 다음 3bytes 는 machine id, 다음 2bytes 는 MongoDB 서버의 프로세스id, 마지막 3bytes 는 순차번호입니다 추가될때마다 값이 높아진다는거지요.

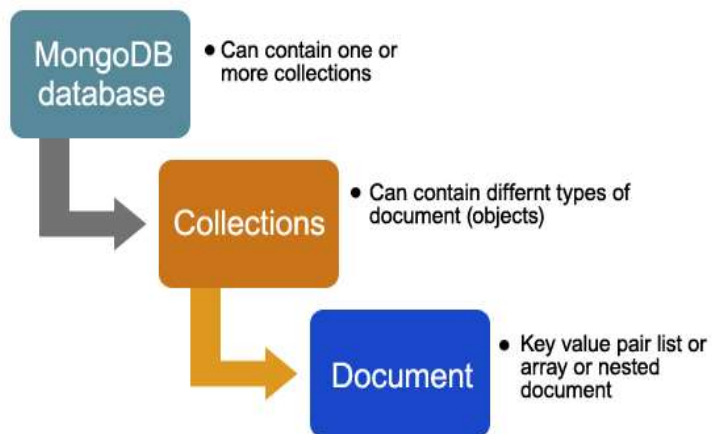
Document는 동적(dynamic)의 schema 를 갖고있습니다. 같은 Collection 안에 있는 Document 끼리 다른 schema 를 갖고 있을 수 있는데요, 쉽게 말하면 서로 다른 데이터 (즉 다른 key) 들을 가지고 있을 수 있습니다.

## Collection?

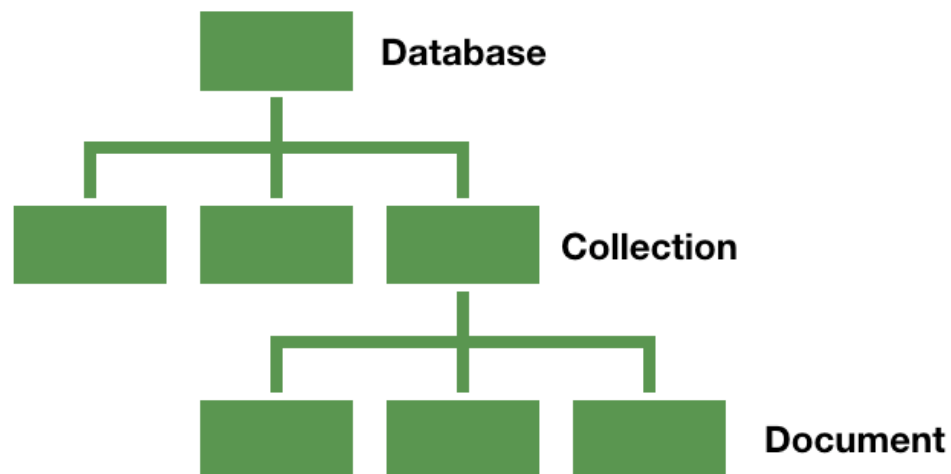
Collection은 MongoDB Document의 그룹입니다. Document들이 Collection 내부에 위치하고 있습니다. RDMS의 table 과 비슷한 개념입니다만 RDMS와 달리 schema 를 따로 가지고 있지않습니다. Document 부분설명에 나와있듯이 각 Document들이 동적인 schema 를 가지고 있으니까요

## Database?

Database는 Collection들의 물리적인 컨테이너입니다. 각 Database는 파일시스템에 여러파일들로 저장됩니다.



<https://cdn.educba.com/academy/wp-content/uploads/2019/04/MongoDB-chart2.jpg>



<https://i.imgur.com/Att4uVC.png>



# A5.9 MongoDB: download



MongoDB Enterprise Server

MongoDB Community Server

MongoDB offers both an Enterprise and Community version of its powerful distributed document database. The community version offers the flexible document model along with ad hoc queries, indexing, and real time aggregation to provide powerful ways to access and analyze your data. As a distributed system you get high availability through built-in replication and failover along with horizontal scalability with native sharding.

The MongoDB Enterprise Server gives you all of this and more. Review the Enterprise Server tab to learn what else is available.

Available Downloads

Version

4.2.11-rc1 (release candidate)

Platform

Windows

Package

msi



Download

Copy Link

<https://www.mongodb.com/try/download/community>

MongoDB 4.2.11 2008R2Plus SSL (64 bit) Service Customi... — □ ×

## Service Configuration

Specify optional settings to configure MongoDB as a service.

☒ Install MongoD as a Service

☒ Run service as Network Service user

☐ Run service as a local or domain user:

Account Domain:

Account Name:

Account Password:

Service Name:

Data Directory:

Log Directory:

< Back    Next >    Cancel

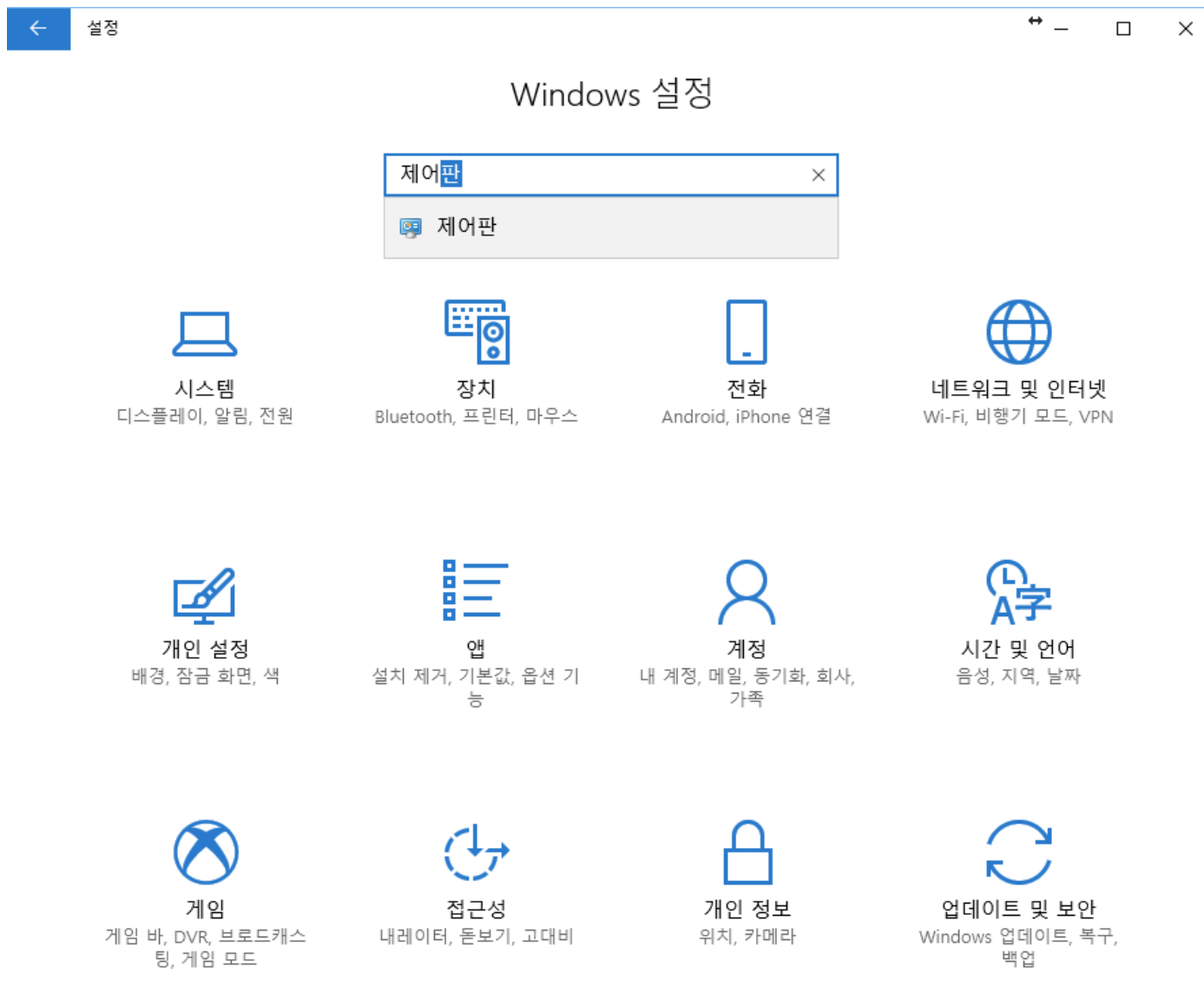


윈도우10: 설정 > 시스템 > 정보

[중요] 시스템 환경변수 : **PATH** 에 경로 추가

**C:\Program Files\MongoDB\Server\4.2\bin**







# A5.9.1 MongoDB install – 3

윈도우10: 설정 > ‘제어판’ 검색 > 모든 제어판 항목에서 ‘시스템’ 선택  
> 고급 시스템 설정

시스템

← → ▾ ▴ [시스템] > 제어판 > 모든 제어판 항목 > 시스템 ▾ [제어판 검색]

제어판 홈

- 장치 관리자
- 원격 설정
- 시스템 보호
- 고급 시스템 설정**

### 컴퓨터에 대한 기본 정보 보기

Windows 버전

Windows 10 Home

© 2017 Microsoft Corporation. All rights reserved.

시스템

프로세서: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.40 GHz

설치된 메모리(RAM): 32.0GB

시스템 종류: 64비트 운영 체제, x64 기반 프로세서

펜 및 터치: 이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.

### 컴퓨터 이름, 도메인 및 작업 그룹 설정

컴퓨터 이름: yish-HCIt

전체 컴퓨터 이름: yish-HCIt

컴퓨터 설명:

작업 그룹: WORKGROUP

### Windows 정품 인증

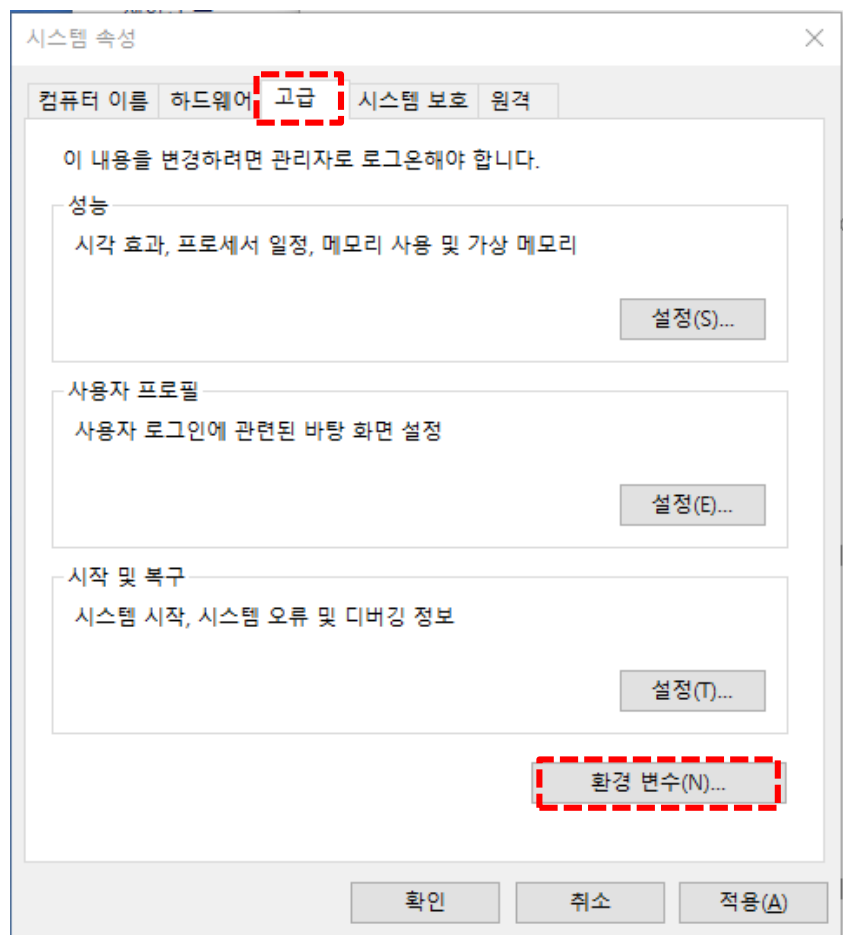
Windows 정품 인증을 받았습니다. [Microsoft 소프트웨어 사용 조건 읽기](#)

제품 ID: 00325-96080-39821-AAOEM

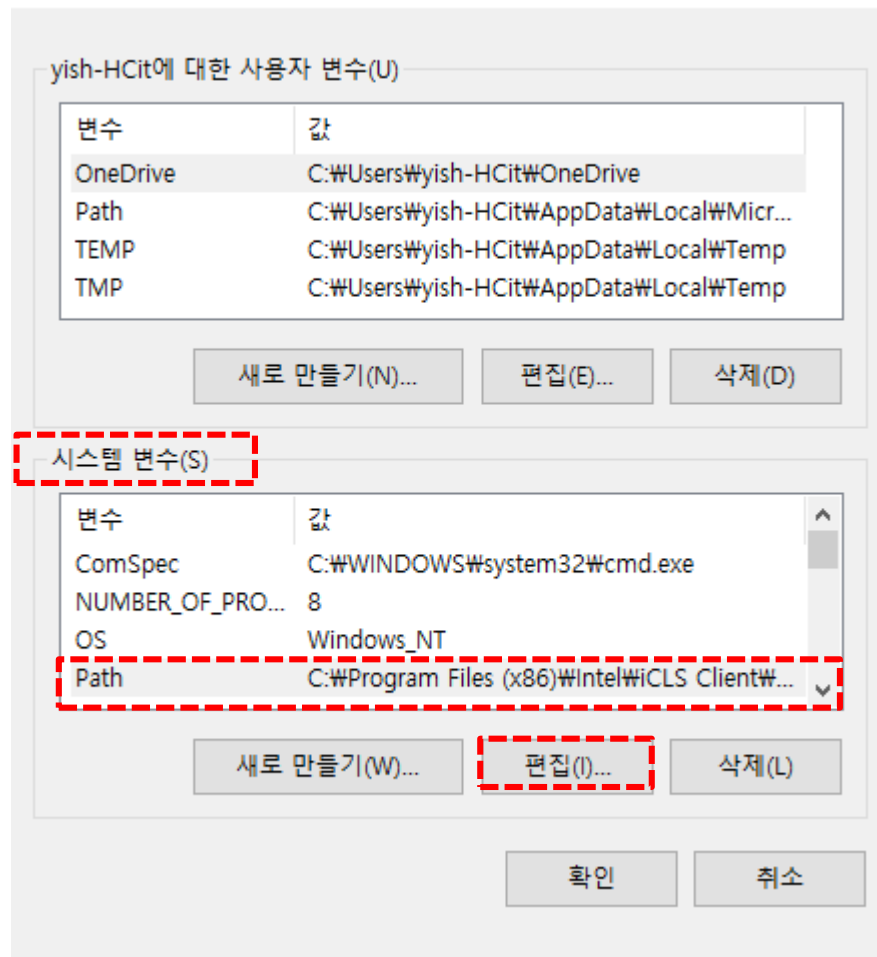
참고 항목

보안 및 유지 관리

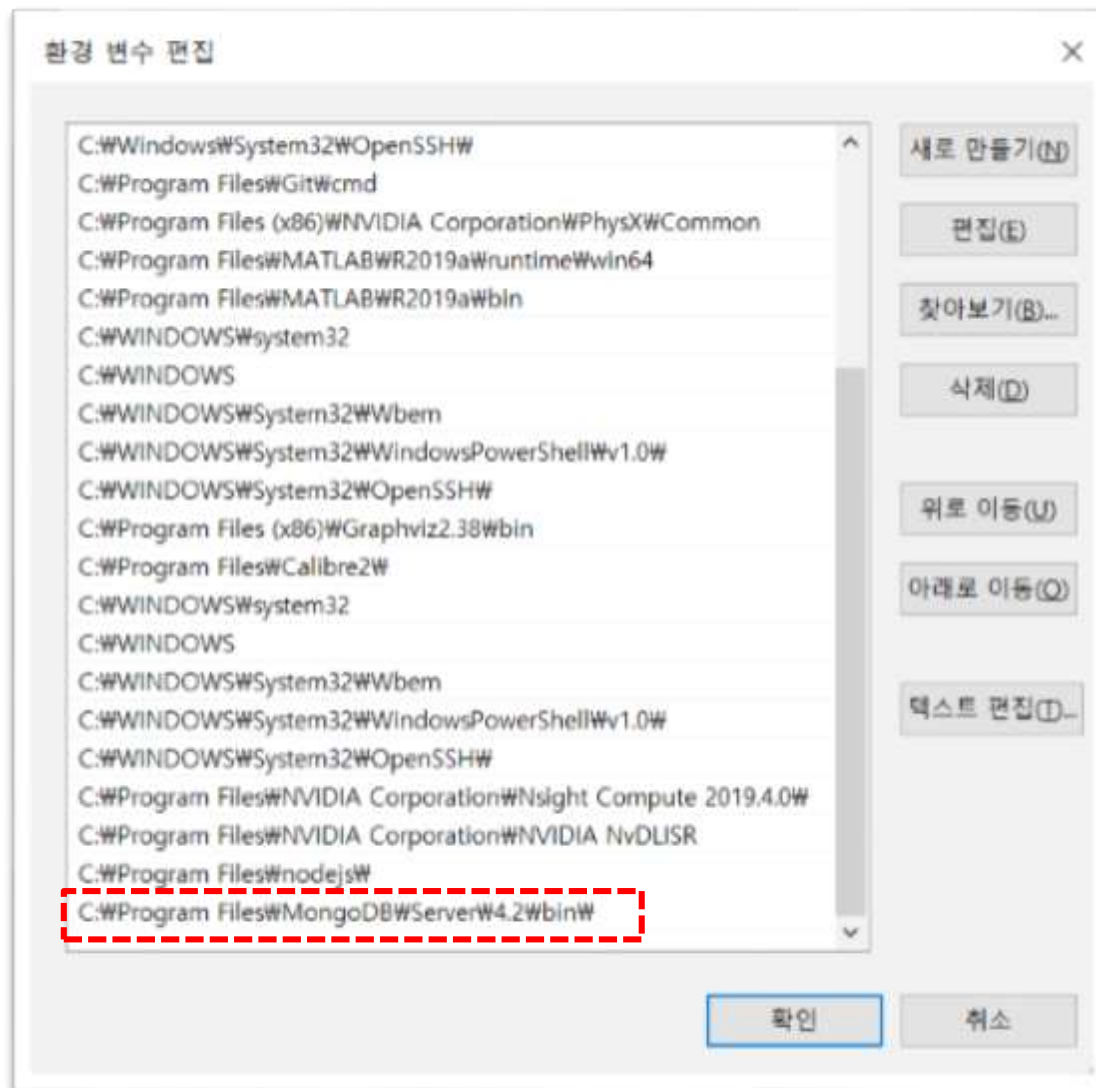
## 환경 변수 설정



### 환경 변수



## 환경 변수 추가





## A5.9.2 MongoDB shell - 1

### 1. Mongo shell 실행

> mongo

C:\ 명령 프롬프트 - mongo

```
D:\mongodb>mongo
MongoDB shell version v4.2.11-rc1
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("862dc45b-18a7-4f26-8006-b28d58799e64") }
MongoDB server version: 4.2.11-rc1
Server has startup warnings:
2020-11-18T09:43:57.884+0900 I CONTROL [initandlisten]
2020-11-18T09:43:57.884+0900 I CONTROL [initandlisten] ** WARNING: Access control
is not enabled for the database.
2020-11-18T09:43:57.884+0900 I CONTROL [initandlisten] **           Read and write
access to data and configuration is unrestricted.
2020-11-18T09:43:57.885+0900 I CONTROL [initandlisten]
---
```

**If, Connect failed... → DB 데몬 실행**

## 2. MongoDB 저장소 만들기 → D drive

- **md mongodb**
- **cd mongodb**
- **dir**
- **md data**
- **md log**
- **dir**

```

명령 프롬프트
D:\mongodb>dir
D 드라이브의 볼륨: DATA
볼륨 일련 번호: 82D1-4852

D:\mongodb 디렉터리

2020-11-18 오전 09:39 <DIR> .
2020-11-18 오전 09:39 <DIR> ..
2020-11-18 오전 10:11 <DIR> data
2020-11-18 오전 09:39 <DIR> log
                                0개 파일
                                4개 디렉터리  2,332,408,369,152 바이트 남음

D:\mongodb>
  
```

사용 **PC** 환경에 맞게 실행 (특히, 경로 지정)

실습실 환경에 맞춰서 **D:**에 **mongodb data** 폴더 지정

## 3. Run MongoDB by using **mongod.exe**

➤ **mongod -dbpath d:\mongodb\data**

명령 프롬프트 - mongod -dbpath d:\mongodb\data

D:\mongodb>md data

D:\mongodb>**mongod -dbpath d:\mongodb\data**

```
2018-01-22T19:27:32.931-0700 I CONTROL [initandlisten] MongoDB starting : pid=18820 port=27017
dbpath=d:\mongodb\data 64-bit host=yish-HCit
2018-01-22T19:27:32.931-0700 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2
008 R2
2018-01-22T19:27:32.932-0700 I CONTROL [initandlisten] db version v3.6.2
2018-01-23T11:27:33.699+0900 I COMMAND [initandlisten] setting featureCompatibilityVersion to
3.6
2018-01-23T11:27:33.706+0900 I STORAGE [initandlisten] createCollection: local.startup_log wit
h generated UUID: 06b3b7cb-62fe-4be5-a929-2a7478650a9b
2018-01-23T11:27:34.211+0900 I FTDC [initandlisten] Initializing full-time diagnostic data
capture with directory 'd:/mongodb/data/diagnostic.data'
2018-01-23T11:27:34.215+0900 I NETWORK [initandlisten] waiting for connections on port 27017
```

사용 **PC** 환경에 맞게 실행 (특히, 경로 지정)



## A5.9.2 MongoDB shell - 4

### 4. Run mongo shell : **mongo.exe** [use new cmd]

➤ **mongo**

Run new cmd

**mongo**

명령 프롬프트 - mongo

```
D:\mongodb>mongo
MongoDB shell version v4.2.11-rc1
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("438383b5-267a-466a-b22e-1bb71bf78985") }
MongoDB server version: 4.2.11-rc1
Server has startup warnings:
2020-11-18T09:43:57.884+0900 I CONTROL [initandlisten]
2020-11-18T09:43:57.884+0900 I CONTROL [initandlisten] ** WARNING: Access control
is not enabled for the database.
2020-11-18T09:43:57.884+0900 I CONTROL [initandlisten] **           Read and write
access to data and configuration is unrestricted.
2020-11-18T09:43:57.885+0900 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and di
splay
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessi
ble to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonit
oring()
---
```





## A5.9.2 MongoDB shell - 5

### 5. mongo shell :

Run new cmd

mongo

show dbs

use local

show  
collections

help

명령 프롬프트 - mongo

```
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
> use local
switched to db local
> show collections
startup_log
> help
```

```
db.help()
db.mycoll.help()
sh.help()
rs.help()
help admin
help connect
help keys
help misc
help mr
```

```
help on db methods
help on collection methods
sharding helpers
replica set helpers
administrative help
connecting to a db help
key shortcuts
misc things to know
mapreduce
```

```
show dbs
show collections
show users
show profile
```

```
show database names
show collections in current database
show users in current database
show most recent system.profile entries with time >=
```

lms

```
show logs
show log [name]
```

```
show the accessible logger names
prints out the last segment of log in memory, 'global'
```

is default

```
use <db_name>
db.foo.find()
db.foo.find( { a : 1 } )
it
```

```
set current database
list objects in collection foo
list objects in foo where a == 1
result of the last line evaluated; use to further ite
```

rate

```
DBQuery.shellBatchSize = x
exit
```

```
set default number of items to display on shell
quit the mongo shell
```



## A5.9.3 MongoDB shell coding

### 1. make my own db (aann) & insert one record (document)

use aa00

show collections

insert record with new collection "user"

**db.user.insert({first:"Redwoods", last:"Yi"})**

show collections

→ "user"

show dbs

**db.user.find()**

```
명령 프롬프트 - mongo
> use aa00
switched to db aa00
> show collections
> db.user.insert({first:"Redwoods", last:"Yi"})
WriteResult({ "nInserted" : 1 })
> show collections
user
> show dbs
aa00      0.000GB
admin     0.000GB
config    0.000GB
local     0.000GB
> _
```



## A5.9.3 MongoDB shell coding

### 2. insert more records with different schema & show records

insert record2

insert record3

show collections

db.user.find()

db.user.find().pretty()

```
명령 프롬프트 - mongo
> db.user.insert({first:"Chaos", last:"Kim"})
WriteResult({"nInserted" : 1})
> db.user.insert({first:"Gildong", last:"Hong"})
WriteResult({"nInserted" : 1})
> show collections
user
> db.user.find()
{ "_id" : ObjectId("5a66b44b9f0d55608f5f7582"), "first" : "Redwoods", "last" : "Yi" }
{ "_id" : ObjectId("5a66b5759f0d55608f5f7583"), "first" : "Chaos", "last" : "Kim" }
{ "_id" : ObjectId("5a66b5869f0d55608f5f7584"), "first" : "Gildong", "last" : "Hong" }
> db.user.find().pretty()
{
  "_id" : ObjectId("5a66b44b9f0d55608f5f7582"),
  "first" : "Redwoods",
  "last" : "Yi"
}
{
  "_id" : ObjectId("5a66b5759f0d55608f5f7583"),
  "first" : "Chaos",
  "last" : "Kim"
}
{
  "_id" : ObjectId("5a66b5869f0d55608f5f7584"),
  "first" : "Gildong",
  "last" : "Hong"
}
>
```

**\_id** 는 12bytes의 hexadecimal 값으로서, 각 document의 유일함(uniqueness)을 제공합니다.

이 값의 첫 4bytes는 현재 timestamp, 다음 3bytes는 machine id, 다음 2bytes는 MongoDB 서버의 프로세스id, 마지막 3bytes는 순차번호입니다.



## A5.9.3 MongoDB shell coding

### 3. insert more records with different schema & show records

insert record4  
with firstName key

`db.user.find()`

`db.user.find().pretty()`

```
명령 프롬프트 - mongo
> db.user.insert({firstName:"Fractal", last:"Park"})
WriteResult({"nInserted": 1})
> db.user.find().pretty()
{
  "_id" : ObjectId("5a66b44b9f0d55608f5f7582"),
  "first" : "Redwoods",
  "last" : "Yi"
}
{
  "_id" : ObjectId("5a66b5759f0d55608f5f7583"),
  "first" : "Chaos",
  "last" : "Kim"
}
{
  "_id" : ObjectId("5a66b5869f0d55608f5f7584"),
  "first" : "Gildong",
  "last" : "Hong"
}
{
  "_id" : ObjectId("5a66b6439f0d55608f5f7585"),
  "firstName" : "Fractal",
  "last" : "Park"
}
>
```

**Dynamic  
schema**

동적스키마

Note that there are two kinds of schemas in JSON.  
Save as

[AAnn\\_mongo\\_schemas.png](#)

## 4. remove one of records (or documents)

remove record3

`db.user.find().pretty()`

```
명령 프롬프트 - mongo
> db.user.remove({last:"Kim"})
writeResult({ nRemoved : 1 })
> db.user.find().pretty()
{
  "_id" : ObjectId("5a66b44b9f0d55608f5f7582"),
  "first" : "Redwoods",
  "last" : "Yi"
}
{
  "_id" : ObjectId("5a66b5869f0d55608f5f7584"),
  "first" : "Gildong",
  "last" : "Hong"
}
{
  "_id" : ObjectId("5a66b6439f0d55608f5f7585"),
  "firstName" : "Fractal",
  "last" : "Park"
}
>
```

## 5. update a record

update record2

`db.user.find().pretty()`

명령 프롬프트 - mongo

```
> db.user.update({last:"Hong"},{$set:{first:"GilDong", age:21}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.user.find().pretty()
{
  "_id" : ObjectId("5a66b44b9f0d55608f5f7582"),
  "first" : "Redwoods",
  "last" : "Yi"
}
{
  "_id" : ObjectId("5a66b5869f0d55608f5f7584"),
  "first" : "GilDong",
  "last" : "Hong",
  "age" : 21
}
{
  "_id" : ObjectId("5a66b6439f0d55608f5f7585"),
  "firstName" : "Fractal",
  "last" : "Park"
}
> _
```

```
db.user.update({last:"Hong"},{$set:{first:"GilDong", age:21}})
```

Note that it is possible to change schema.  
Save as

[AAnn\\_mongo\\_update.png](#)





# Node.js



+

# MongoDB

