

# **MongoDB 실습**

---

## **Database Programming**

 MongoDB 소개

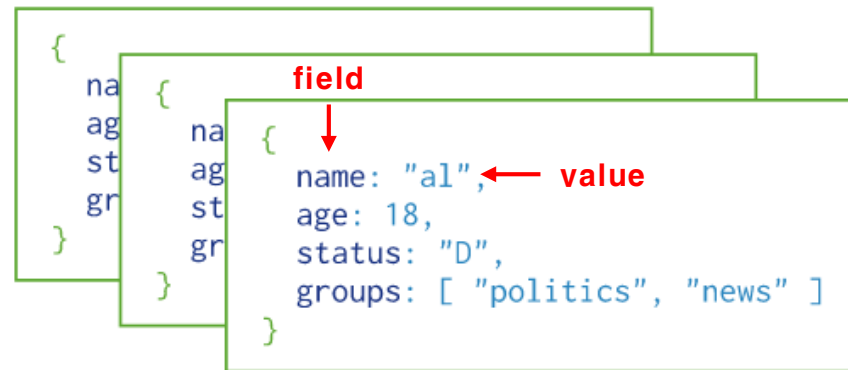
 MongoDB 설치

 MongoDB 실행

 MongoDB 명령어

 MongoDB 연산자

- MongoDB
  - 크로스 플랫폼 문서를 지향하는 NoSQL 데이터베이스 시스템
  - 데이터를 필드(field)와 값(value)의 쌍으로 구성된 JSON 객체와 유사한 구조인 BSON(Binary JSON) 문서로 저장함
  - 문서(데이터)를 컬렉션에 저장하고 데이터베이스에 하나 이상의 컬렉션을 저장함
  - 무료 오픈 소스 소프트웨어



[그림] Collection 구조<sup>[1]</sup>

[1] <https://docs.mongodb.com/manual/core/databases-and-collections/>

## 1) <https://www.mongodb.com/try/download/community> 접속

The screenshot shows the MongoDB website's download page for the Community Server. The top navigation bar includes the MongoDB logo, links for Products, Solutions, Resources, Company, and Pricing, a search icon, a Sign In link, and a prominent green 'Try Free' button. The left sidebar lists various MongoDB offerings: MongoDB Atlas, MongoDB Enterprise Advanced, MongoDB Community Edition, MongoDB Community Server (which is highlighted with a green vertical bar), MongoDB Community Kubernetes Operator, Tools, and Mobile & Edge. The main content area is titled 'MONGODB COMMUNITY SERVER' and 'MongoDB Community Server Download'. It features a descriptive paragraph about the Community version's flexibility and support for ad-hoc queries, secondary indexing, and real-time aggregations. Below this, it mentions that the database is also available as a fully-managed service with MongoDB Atlas, listing advanced functionalities like auto-scaling, serverless instances, full-text search, and multi-region data distribution. A blue link offers a 'free, highly-available 512 MB cluster' for a trial. At the bottom, there are two dropdown menus: 'Version' set to '6.0.3 (current)' and 'Platform' set to 'Windows'.

MongoDB

Products Solutions Resources Company Pricing

Search Sign In Try Free

MongoDB Atlas

MongoDB Enterprise Advanced

MongoDB Community Edition

**MongoDB Community Server**

MongoDB Community Kubernetes Operator

Tools

Mobile & Edge

MONGODB COMMUNITY SERVER

## MongoDB Community Server Download

The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data.

The database is also offered as a fully-managed service with [MongoDB Atlas](#). Get access to advanced functionality such as auto-scaling, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

[Give it a try with a free, highly-available 512 MB cluster.](#)

Version  
6.0.3 (current)

Platform  
Windows

## 2) "6.0.3" 버전, 자신의 운영체제에 맞는 플랫폼, "msi" 패키지 선택 후 다운로드 클릭

The screenshot shows the MongoDB download page. The left sidebar lists various MongoDB products, with 'MongoDB Community Server' selected. The main content area describes the database as a fully-managed service with MongoDB Atlas. Below this, a section titled 'Give it a try with a free, highly-available 512 MB cluster.' contains a configuration table. This table is highlighted with a red border and shows the following selections: Version 6.0.3 (current), Platform Windows, and Package msi. Below the table is a green 'Download' button, also highlighted with a red border. To the right of the button is a 'More Options' link with three dots.

Version
6.0.3 (current)

Platform
Windows

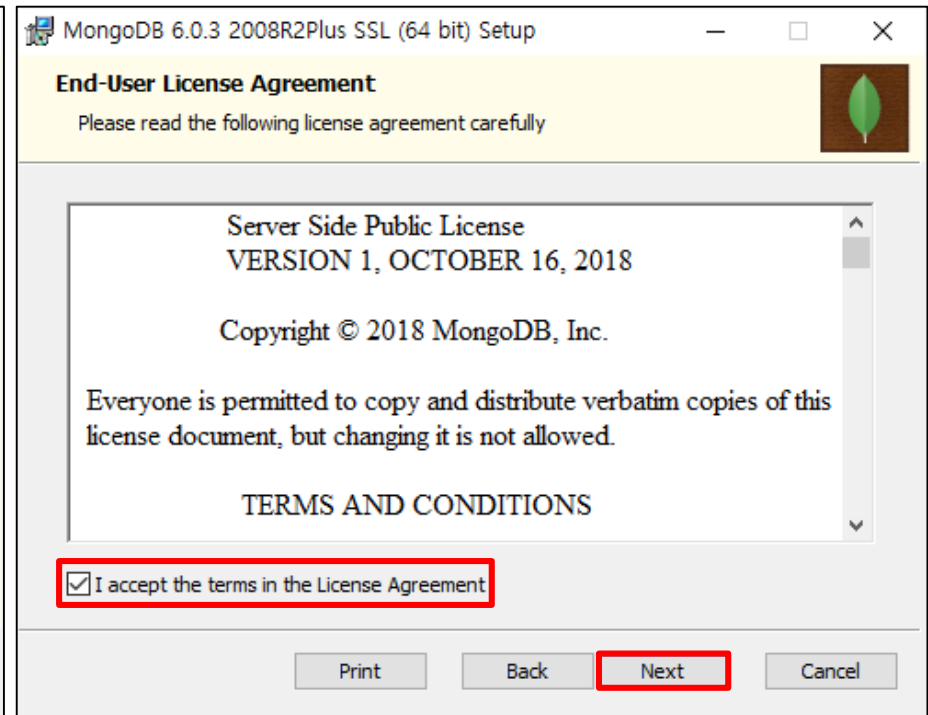
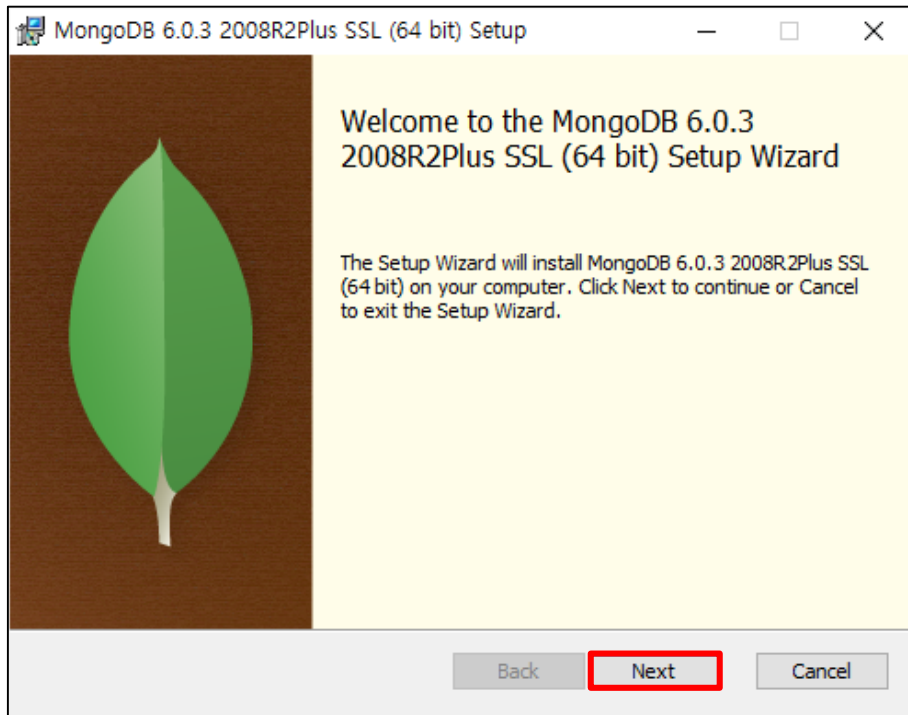
Package
msi

[Download](#)

[More Options](#) ...

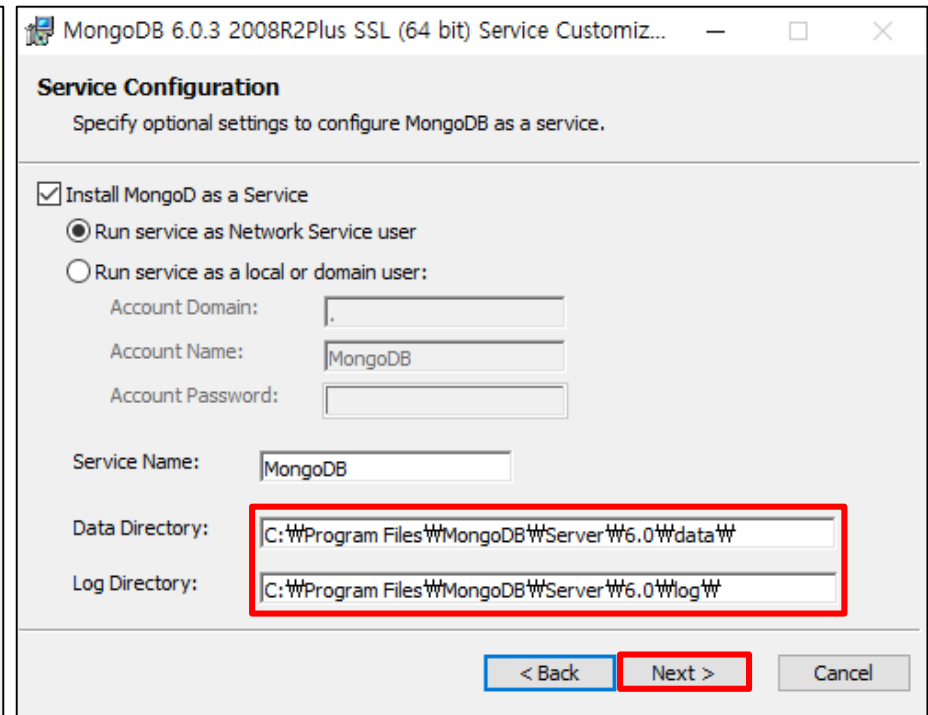
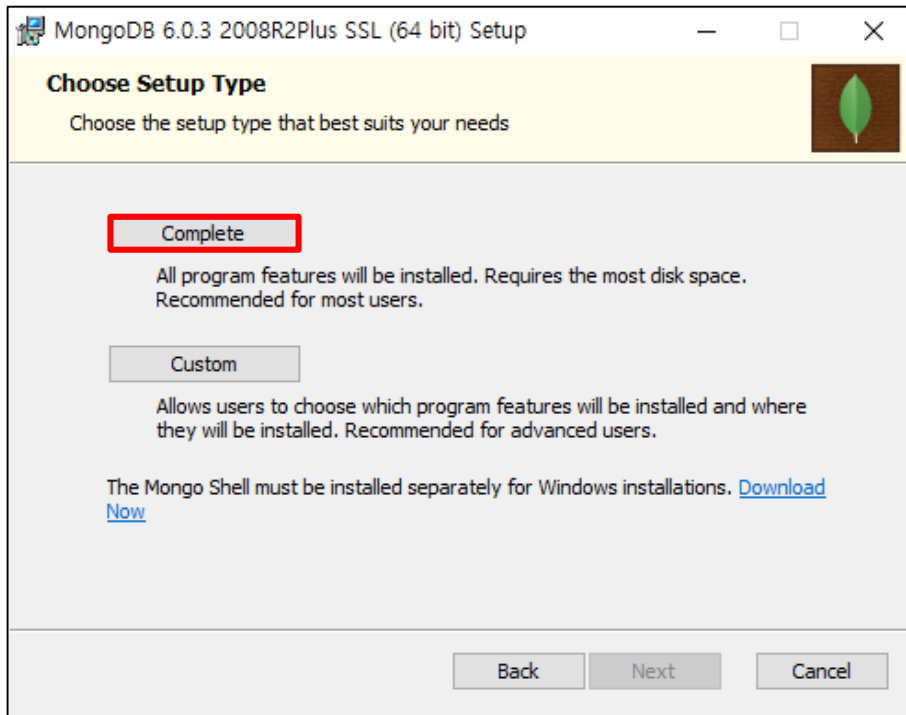
3) 다운로드 받은 설치 파일 실행 → Next 클릭

4) 라이선스 동의 → Next 클릭



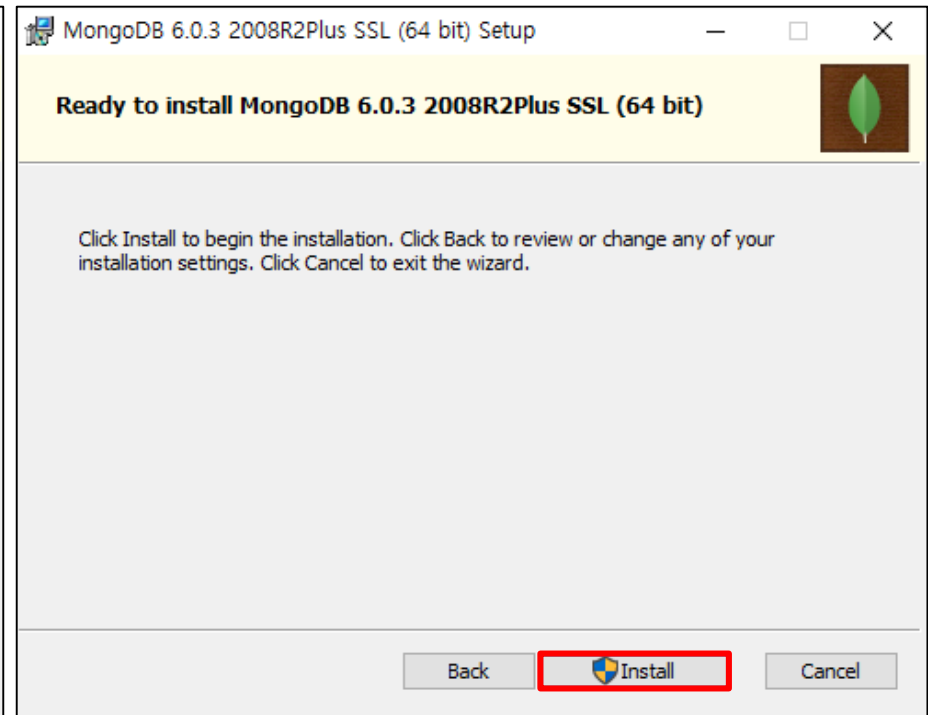
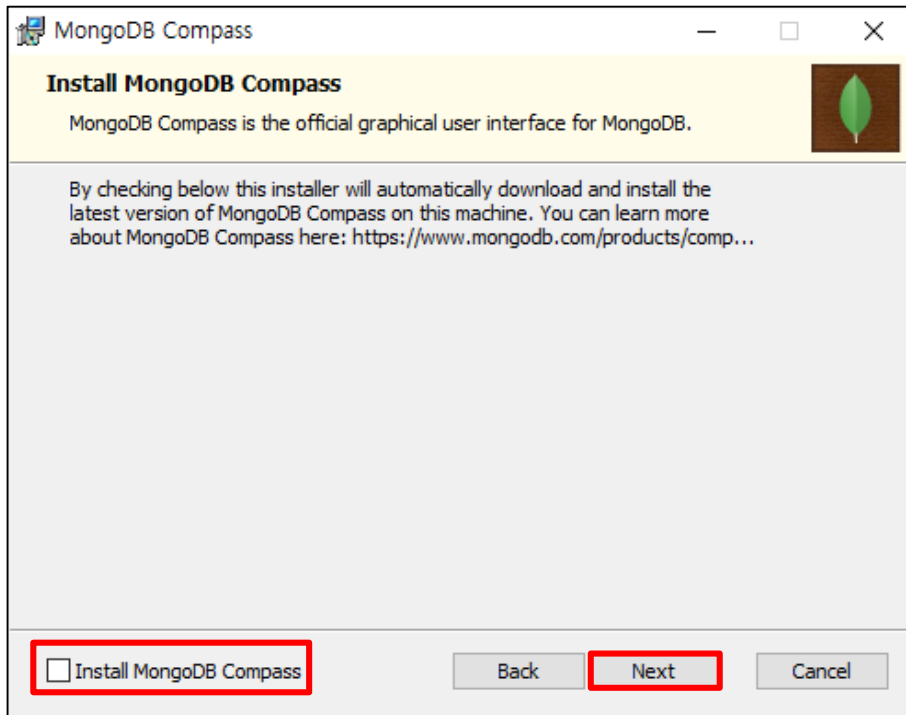
5) Complete 클릭

6) 설치 경로 확인 → Next 클릭



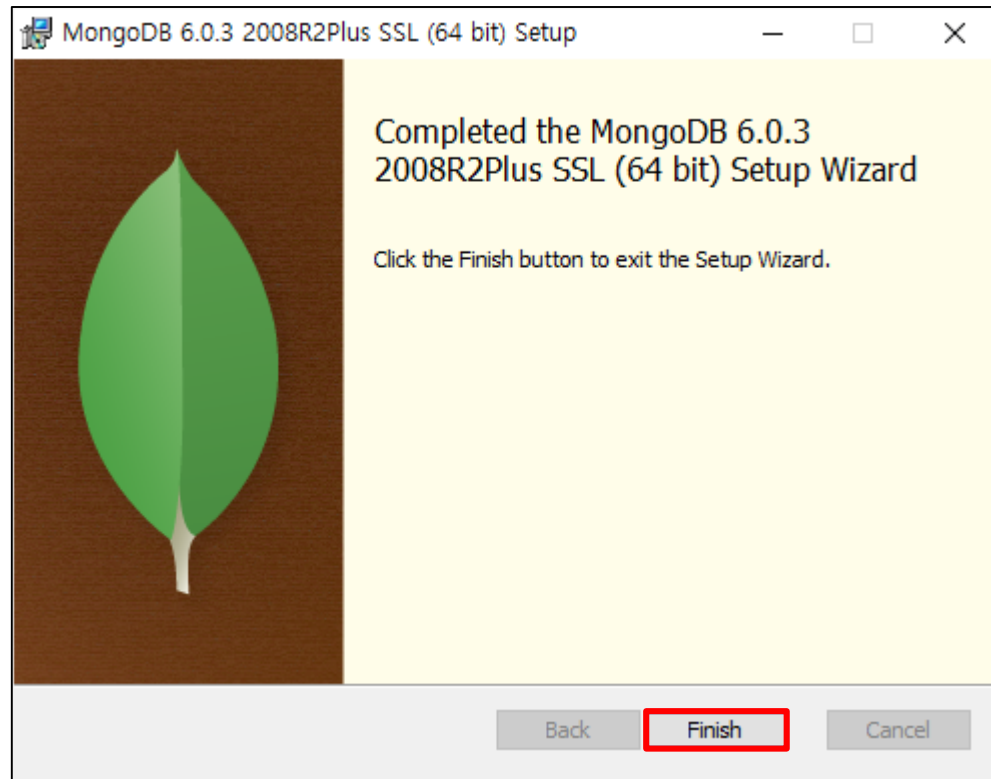
7) Install MongoDB Compass 체크 해제 → Next 클릭

8) Install 클릭

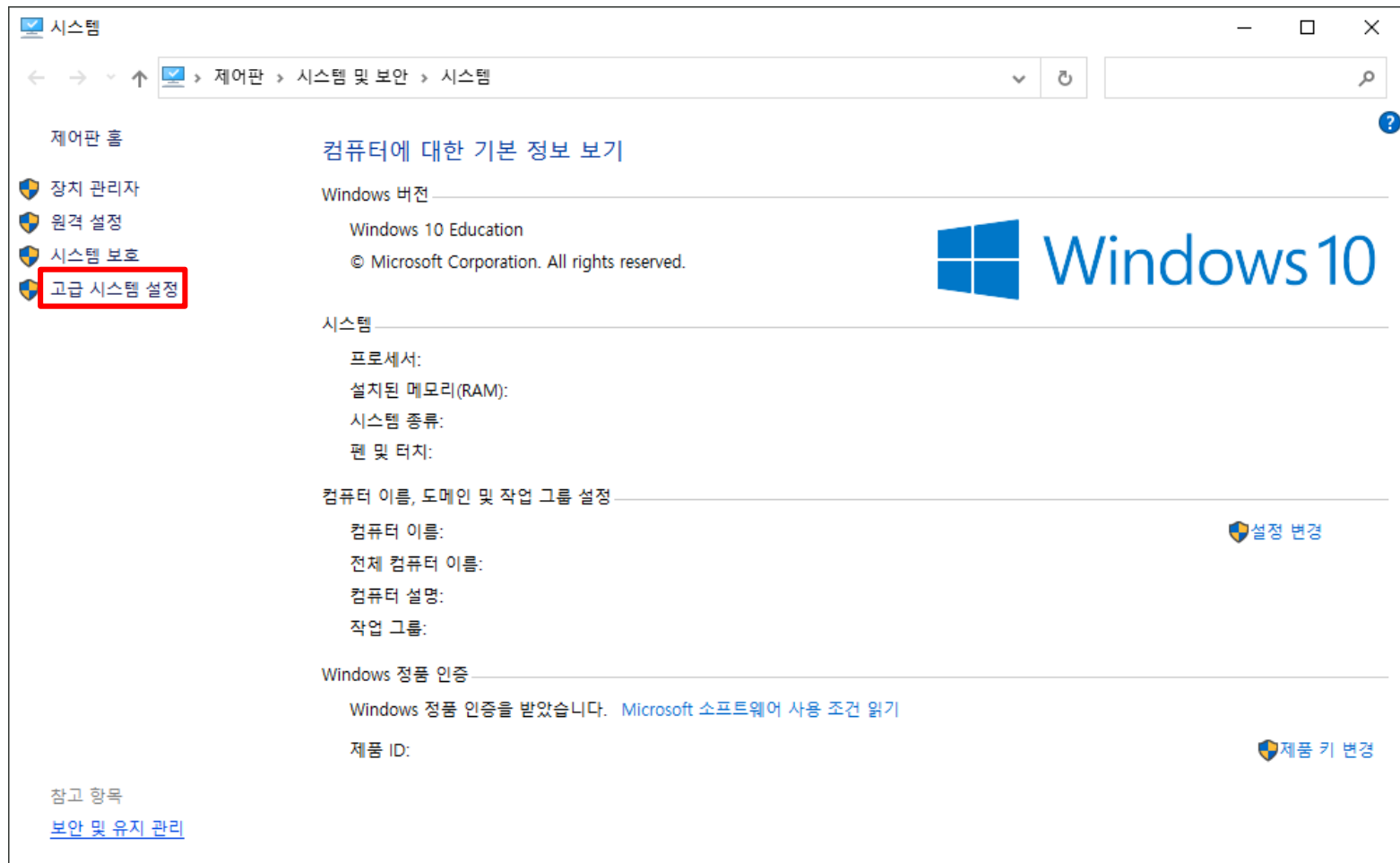




## 9) Finish 클릭

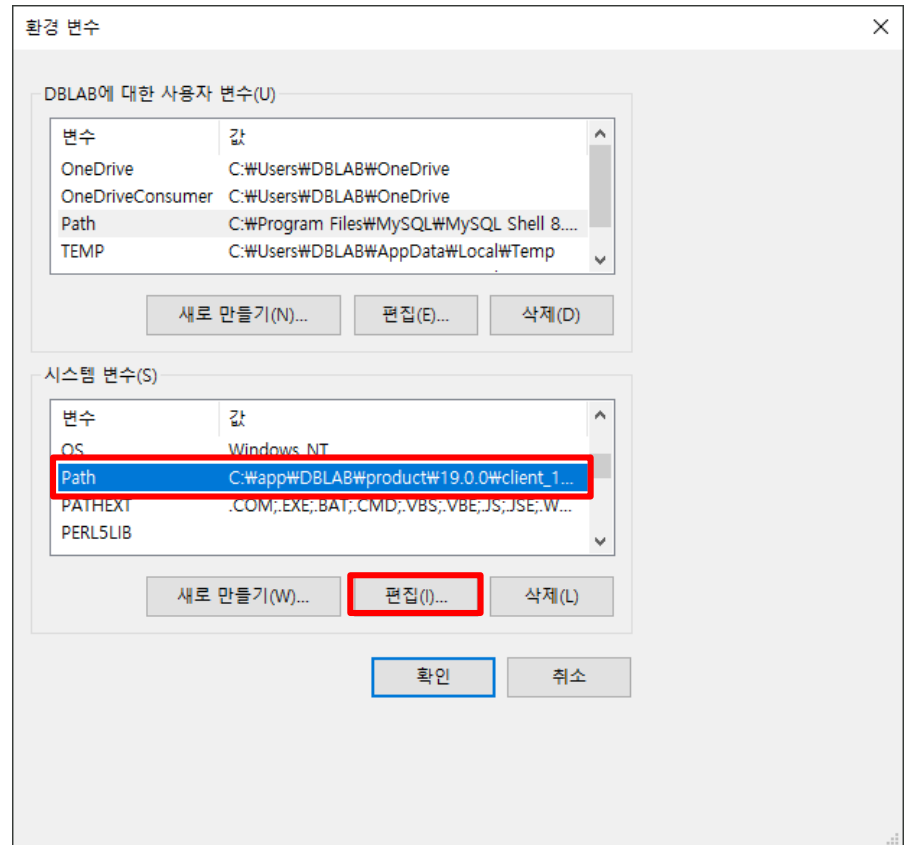
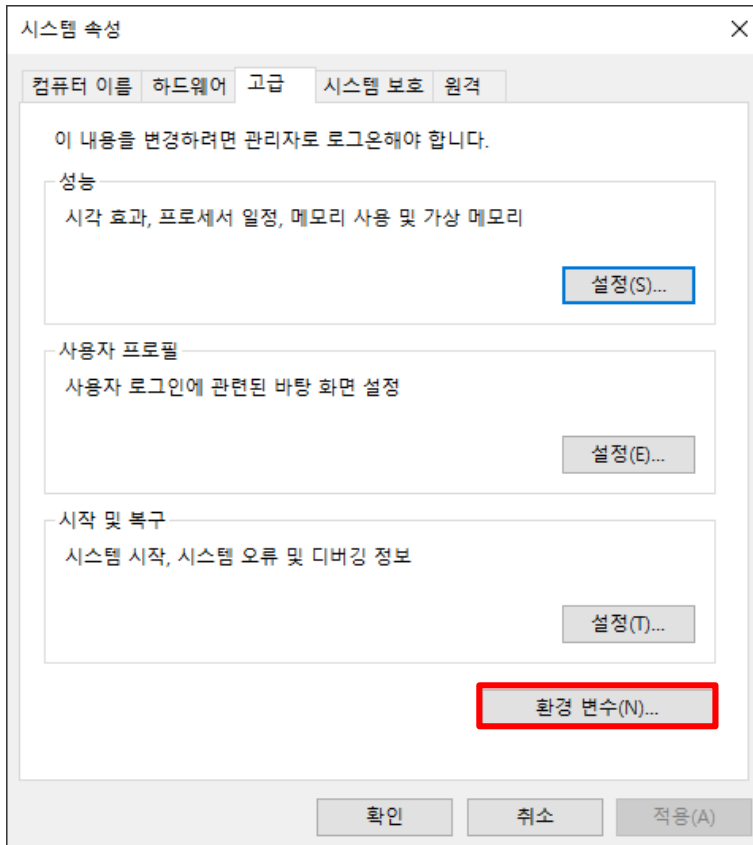


## 10) 내 PC 우클릭 → 속성 → 고급 시스템 설정 클릭



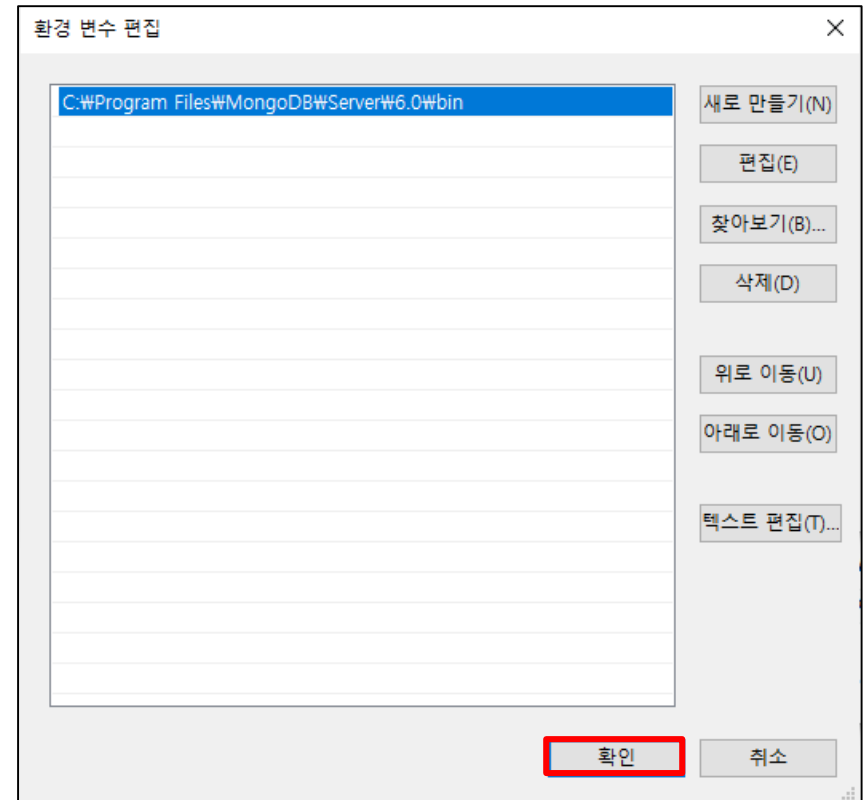
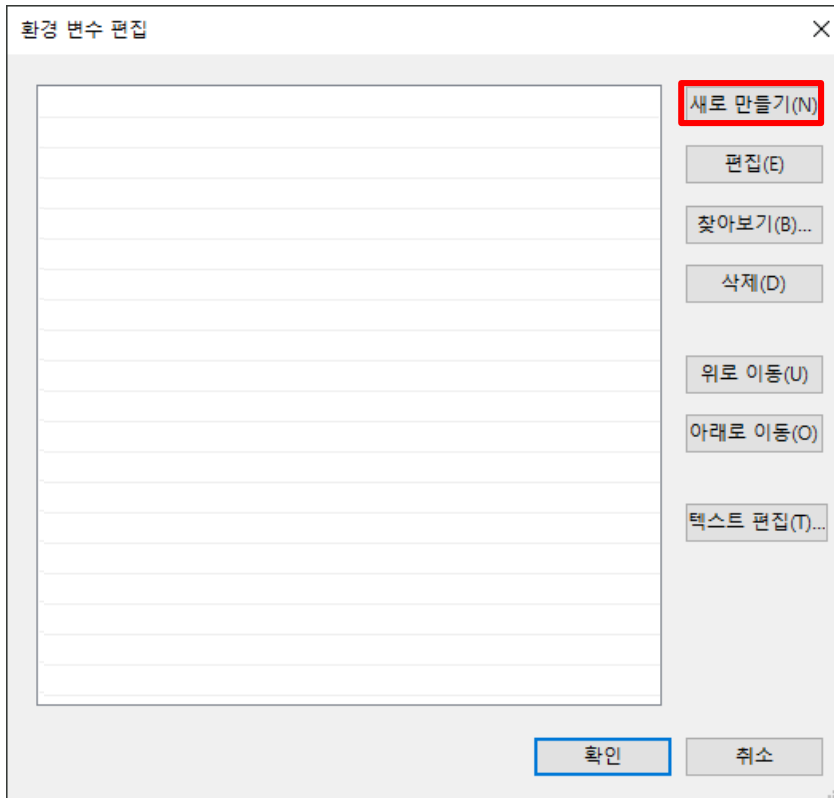
## 11) 환경 변수 클릭

## 12) 시스템 변수 → Path 선택 → 편집 클릭



13) 새로 만들기 클릭

14) C:\Program Files\MongoDB\Server\6.0\bin 입력 → 확인 클릭



## 15) cmd 실행 → mongod --version 입력하여 설치 확인



```
cmd 명령 프롬프트
C:\#Users#DBLAB>mongod --version
db version v6.0.3
Build Info: {
  "version": "6.0.3",
  "gitVersion": "f803681c3ae19817d31958965850193de067c516",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
C:\#Users#DBLAB>
```

## 1) <https://docs.mongodb.com/mongodb-shell/install/> 접속

The screenshot shows the MongoDB documentation website for the MongoDB Shell installation. The page has a top navigation bar with links for Products, Solutions, Resources, Company, and Pricing. A search icon and a 'Sign In' link are on the right, along with a green 'Try Free' button. The left sidebar contains a 'MongoDB Documentation' section with a 'MongoDB Shell' subsection. Under 'MongoDB Shell', 'Install mongosh' is highlighted. Below it are links for 'Connect to a Deployment', 'Configure mongosh', 'Run Commands', 'Perform CRUD Operations', 'Run Aggregation Pipelines', 'Client-Side Field Level Encryption', 'Write Scripts', 'Snippets', 'Reference', and 'mongosh Help'. The main content area shows the breadcrumb 'Docs Home → View & Analyze Data → MongoDB Shell' and the title 'Install mongosh'. Below the title is a 'Prerequisites' section stating that a MongoDB deployment is needed, with links to 'MongoDB Atlas' and 'Install MongoDB'. A 'Supported MongoDB Versions' section mentions version 4.0 or greater. The 'Procedure' section instructs users to select a tab for their operating system: Windows, macOS, or Linux. The 'Windows' tab is active. A 'NOTE' box states that on Windows, mongosh preferences and configuration options are stored in the %APPDATA%/mongodb/mongosh directory. On the right, a 'On this page' sidebar lists 'Prerequisites', 'Procedure', and 'Next Steps'. A 'Share Feedback' button is at the bottom right.

MongoDB. Products Solutions Resources Company Pricing

Search Sign In Try Free

MongoDB Documentation

← Back To View & Analyze Data

MongoDB Shell

Install mongosh

Connect to a Deployment

► Configure mongosh

Run Commands

► Perform CRUD Operations

Run Aggregation Pipelines

Client-Side Field Level Encryption

► Write Scripts

► Snippets

► Reference

mongosh Help

Docs Home → View & Analyze Data → MongoDB Shell

## Install mongosh

### Prerequisites

To use the MongoDB Shell, you must have a MongoDB deployment to connect to.

- For a free cloud-hosted deployment, you can use [MongoDB Atlas](#).
- To learn how to run a local MongoDB deployment, see [Install MongoDB](#).

### Supported MongoDB Versions

You can use the MongoDB Shell to connect to MongoDB version 4.0 or greater.

### Procedure

Select the appropriate tab for your operating system:

Windows macOS Linux

**NOTE**

On Windows, `mongosh` preferences and configuration options are stored in the `%APPDATA%/mongodb/mongosh` directory.

On this page

- Prerequisites
- Procedure
- Next Steps

Share Feedback

## 2) "MongoDB Download Center" 클릭

The screenshot shows the MongoDB documentation page for installing the MongoDB Shell (mongosh) on Windows. The page is titled "MongoDB Shell" and includes a sidebar with navigation links. The main content area is titled "Procedure" and includes a "NOTE" box stating that mongosh preferences and configuration options are stored in the %APPDATA%/mongodb/mongosh directory. The "Install from MSI" section lists five steps: 1. Open the MongoDB Shell download page. 2. In the Platform dropdown, select Windows 64-bit (8.1+) (MSI). 3. Click Download. 4. Double-click the installer file. 5. Follow the prompts to install mongosh. The link "MongoDB Download Center" in step 1 is highlighted with a red box. The "Install from .zip File" section is partially visible at the bottom.

**MongoDB** Products Solutions Resources Company Pricing Q Sign In Try Free

**MongoDB Documentation**

- ← Back To View & Analyze Data
- MongoDB Shell**
  - Install mongosh**
  - Connect to a Deployment
  - ▶ Configure mongosh
  - Run Commands
  - ▶ Perform CRUD Operations
  - Run Aggregation Pipelines
  - Client-Side Field Level Encryption
  - ▶ Write Scripts
  - ▶ Snippets
  - ▶ Reference
  - mongosh Help

You can use the MongoDB Shell to connect to MongoDB version 4.0 or greater.

### Procedure

Select the appropriate tab for your operating system:

**Windows** macOS Linux

**NOTE**

On Windows, mongosh preferences and configuration options are stored in the %APPDATA%/mongodb/mongosh directory.

#### Install from MSI

- 1 Open the MongoDB Shell download page.  
Open the **MongoDB Download Center**.
- 2 In the **Platform** dropdown, select **Windows 64-bit (8.1+) (MSI)**
- 3 Click **Download**.
- 4 Double-click the installer file.
- 5 Follow the prompts to install mongosh.

#### Install from .zip File

Share Feedback

## 3) "1.6.1" 버전, 자신의 운영체제에 맞는 플랫폼, "msi" 패키지 선택 후 다운로드 클릭

MongoDB Atlas

MongoDB Enterprise Advanced

MongoDB Community Edition

Tools

**MongoDB Shell**

MongoDB Compass

Atlas CLI

Atlas Kubernetes Operator

MongoDB CLI for Cloud Manager and Ops Manager

MongoDB Cluster-to-Cluster Sync

MongoDB Database Tools

### MongoDB Shell Download

MongoDB Shell is the quickest way to connect to (and work with) MongoDB. Easily query data, configure settings, and execute other actions with this modern, extensible command-line interface – replete with syntax highlighting, intelligent autocomplete, contextual help, and error messages.

**Note:** MongoDB Shell is an open source (Apache 2.0), standalone product developed separately from the MongoDB Server.

[Learn more](#)

Version

1.6.1

▼

Platform

Windows 64-bit (8.1+) (MSI)

▼

Package

msi

▼

Download

⬇

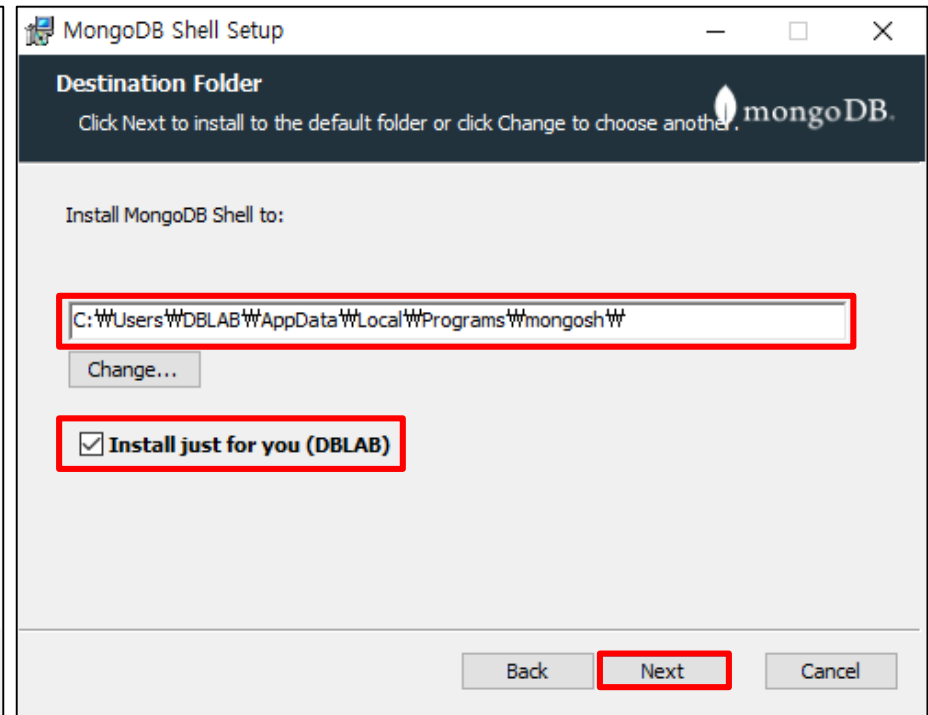
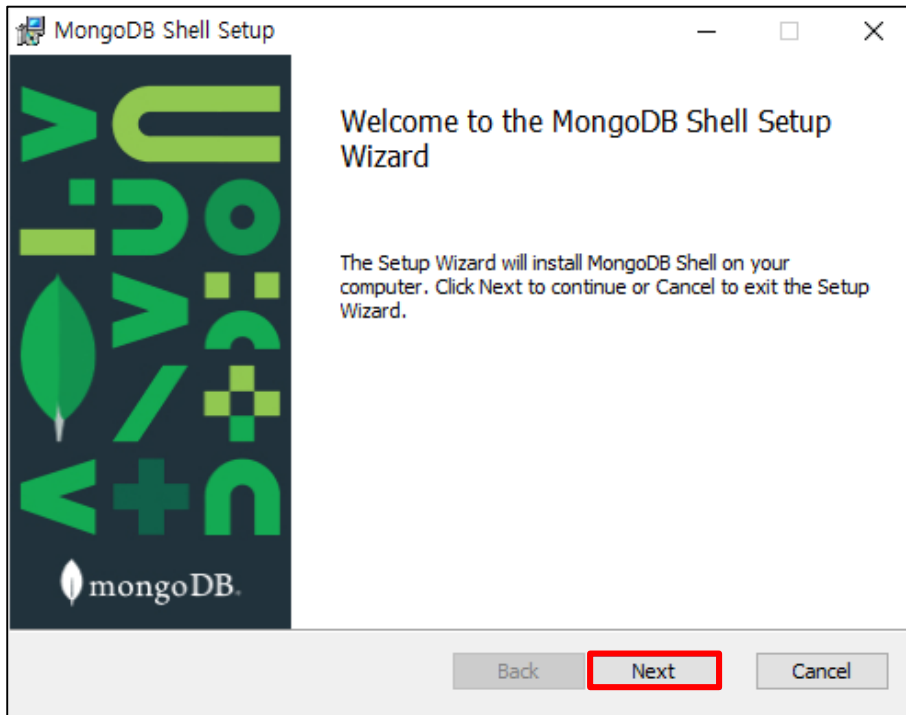
More Options

⋮



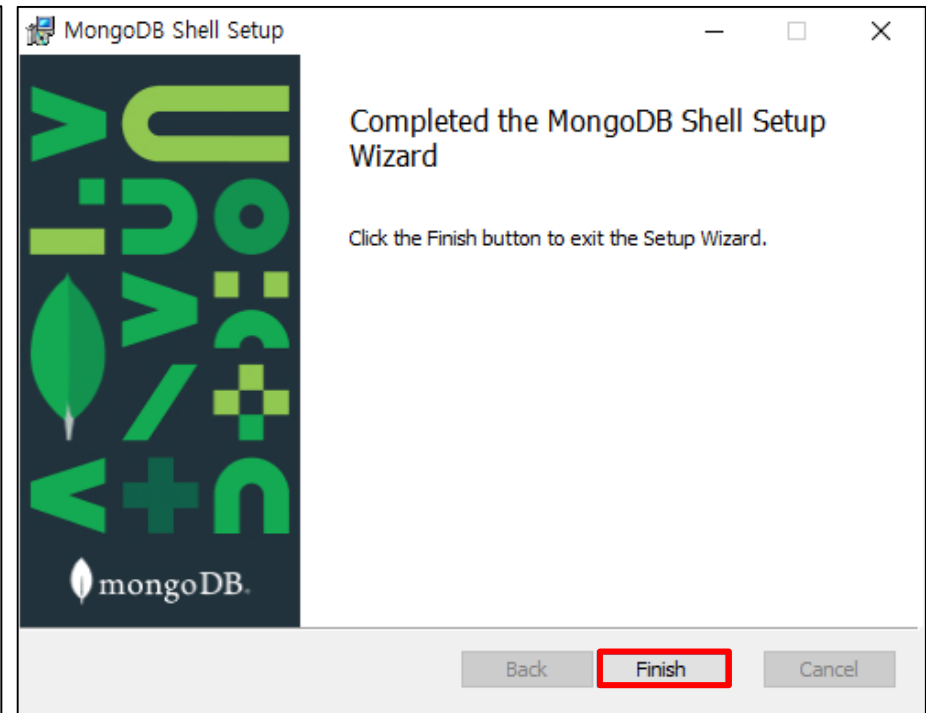
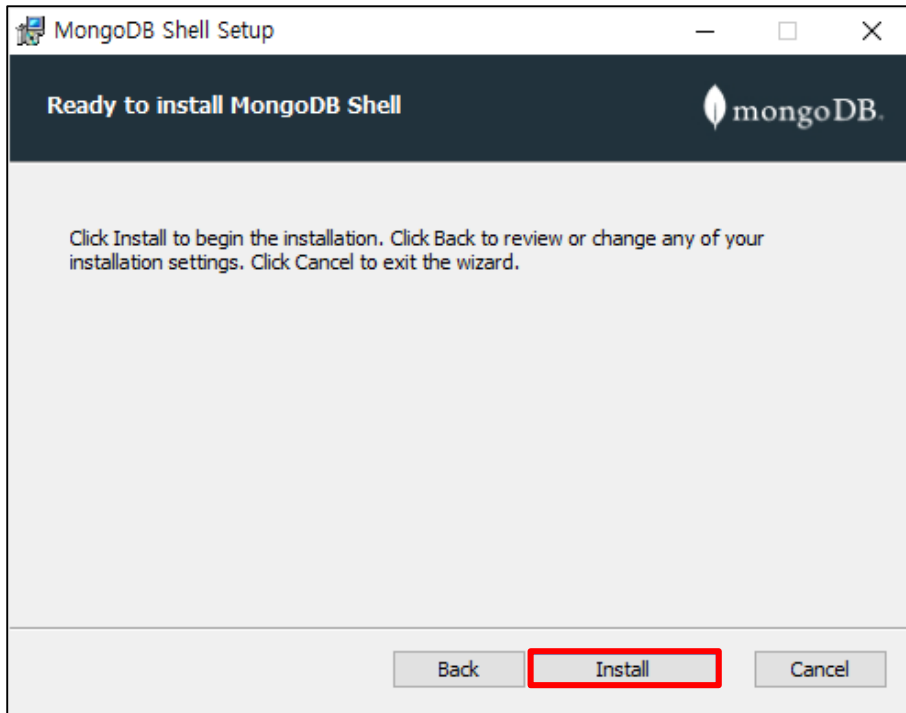
4) 다운로드 받은 설치 파일 실행 → Next 클릭

5) 설치 경로 확인 → Install just for you 체크 → Next 클릭



6) Install 클릭

7) 설치 완료 후 Finish 클릭



## 1) cmd창에 "mongosh" 입력 → "27017" 포트로 실행됨을 확인

```
C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

C:\Users\#DBLAB> mongosh
Current Mongosh Log ID: 6395d7547bc451c3d9e2237a
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.1
test>
Using Mongosh:      1.6.1

For mongosh info see: https://docs.mongodb.com/mongodbd-shell/

-----
The server generated these startup warnings when booting
2022-12-11T21:21:31.431+09:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible 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.disableFreeMonitoring()

test>
```

## 2) "show dbs" 입력

```
C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
test> show dbs
admin      40.00 KiB
config     60.00 KiB
local      80.00 KiB
testdb     144.00 KiB
test> _
```

- 데이터베이스 생성
  - 명령어
    - use <생성할 데이터베이스 이름>
  - 예제) testdb 라는 이름의 데이터베이스 생성
    - use testdb

```
test> use testdb
switched to db testdb
testdb>
```

- 현재 사용하고 있는 데이터베이스 조회
  - 명령어
    - db

```
testdb> db
testdb
```

- Collection 생성

- Collection

- 데이터베이스 내에 실제로 사용되는 도큐먼트

- 명령어

- db.createCollection( “<생성할 collection 이름>” )

- 예제) customers 라는 이름의 collection 생성

- db.createCollection( “customers” )

```
testdb> db.createCollection("customers")
{ ok: 1 }
testdb>
```

- 데이터 입력

- 명령어

- db.<collection 이름>.insertOne(<생성할 데이터>)
    - db.컬렉션명.insertOne({필드명:"값", ...})

- 예제) customers 라는 collection 안에 이름은 gildong, 나이는 22인 고객의 데이터를 생성

```
testdb> db.customers.insertOne({name:"gildong", age: 22})
{
  acknowledged: true,
  insertedId: ObjectId("61af1d91a8497cd40f6ce5ad")
}
testdb>
```



## ■ 데이터 조회

### • 명령어

- `db.<collection 이름>.find`  
(<검색 조건>)
- `db.컬렉션명.find`  
({필드명 : "조건값", ...})
- `db.customers.find()` 를 사용  
하는 경우 전체 내용 검색

- 예제) customers 라는 collection  
안에 이름이 gildong인 데이터 검색

```
testdb> db.customers.find()
[
  {
    _id: ObjectId("61af19d3a8497cd40f6ce5a6"),
    name: 'gildong',
    age: 22
  },
  {
    _id: ObjectId("61af1a78a8497cd40f6ce5a7"),
    name: 'cheolsu',
    age: 33
  },
  {
    _id: ObjectId("61af1a7ba8497cd40f6ce5a8"),
    name: 'yeonghui',
    age: 32
  }
]
testdb> db.customers.find({name: "gildong"})
[
  {
    _id: ObjectId("61af19d3a8497cd40f6ce5a6"),
    name: 'gildong',
    age: 22
  }
]
testdb>
```

## ■ 데이터 갱신

### • 명령어

- db.<collection 이름>.updateOne

(<갱신할 데이터 선택>, <갱신할 데이터 입력>)

- db.컬렉션명.updateOne

({필드명:"조건값", ...}, {\$set: {필드명:"변경값", ...}})

```
testdb> db.customers.updateOne({name:"gildong"}, {$set: {name:"hana"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
testdb> db.customers.find()
[
  { _id: ObjectId("61af1d91a8497cd40f6ce5ad"), name: 'hana', age: 22 },
  {
    _id: ObjectId("61af1dd4a8497cd40f6ce5ae"),
    name: 'cheolsu',
    age: 33
  },
  {
    _id: ObjectId("61af1dd4a8497cd40f6ce5af"),
    name: 'yeonghui',
    age: 32
  }
]
```

## ■ 데이터 삭제

### • 명령어

- `db.<collection 이름>.deleteOne(<삭제할 데이터 선택>)`
- `db.컬렉션명.deleteOne({필드명:"조건값", ...})`

- 예제) customers 라는 collection 안에 이름이 hana인 데이터를 삭제

```
testdb> db.customers.deleteOne({name:"hana"});
{ acknowledged: true, deletedCount: 1 }
testdb> db.customers.find()
[
  {
    _id: ObjectId("61af1dd4a8497cd40f6ce5ae"),
    name: 'cheolsu',
    age: 33
  },
  {
    _id: ObjectId("61af1dd4a8497cd40f6ce5af"),
    name: 'yeonghui',
    age: 32
  }
]
testdb>
```

# MongoDB 명령어 모음(1/2)

27/29

Name	Methods	Description	Example
insert	insertOne	단일의 도큐먼트를 collection에 insert	<pre>db.collection.insertOne(   &lt;document&gt;,   {     writeConcern: &lt;document&gt;   } )</pre>
	insertMany	다수의 도큐먼트를 collection에 insert	<pre>db.collection.insertMany(   [ &lt;document 1&gt;, &lt;document 2&gt;, ... ],   {     writeConcern: &lt;document&gt;,     ordered: &lt;boolean&gt;   } )</pre>
update	updateOne	특정 조건에 맞는 다수의 도큐먼트가 있더라도 하나의 도큐먼트만 update	<pre>db.collection.updateOne(   &lt;filter&gt;, &lt;update&gt;,   {     upsert: &lt;boolean&gt;,     writeConcern: &lt;document&gt;,     collation: &lt;document&gt;,     arrayFilters: [ &lt;filterdocument1&gt;, ... ],     hint: &lt;document string&gt;   } )</pre>
	updateMany	특정 조건에 맞는 모든 도큐먼트를 update	<pre>db.collection.updateMany(   &lt;filter&gt;, &lt;update&gt;,   {     upsert: &lt;boolean&gt;,     writeConcern: &lt;document&gt;,     collation: &lt;document&gt;,     arrayFilters: [ &lt;filterdocument1&gt;, ... ],     hint: &lt;document string&gt;   } )</pre>

Name	Methods	Description	Example
update	replaceOne	특정 조건에 맞는 다수의 도큐먼트가 있더라도 하나의 도큐먼트만 replace	<pre>db.collection.replaceOne(   &lt;filter&gt;, &lt;replacement&gt;,   {     upsert: &lt;boolean&gt;,     writeConcern: &lt;document&gt;,     collation: &lt;document&gt;,     hint: &lt;document string&gt;   } )</pre>
delete	deleteOne	특정 조건에 맞는 다수의 도큐먼트가 있더라도 하나의 도큐먼트만 delete	<pre>db.collection.deleteOne(   &lt;filter&gt;,   {     writeConcern: &lt;document&gt;,     collation: &lt;document&gt;,     hint: &lt;document string&gt;   } )</pre>
	deleteMany	특정 조건에 맞는 모든 도큐먼트를 delete	<pre>db.collection.deleteMany(   &lt;filter&gt;,   {     writeConcern: &lt;document&gt;,     collation: &lt;document&gt;   } )</pre>

## ■ 비교 연산자

Name	Description	Example
\$eq(equals)	주어진 값과 일치하는 값	Syntax: ( field: { \$eq: value } )
\$gt(greater than)	주어진 값보다 큰 값	Syntax: ( field: { \$gt: value } )
\$gte(greater than or equals)	주어진 값보다 크거나 같은 값	Syntax: ( field: { \$gte: value } )
\$in	주어진 배열 안에 속하는 값	Syntax: ( field: { \$in: [ value1, ... , valueN ] } )
\$lt(less than)	주어진 값보다 작은 값	Syntax: ( field: { \$lt: value } )
\$lte(less than or equals)	주어진 값보다 작거나 같은 값	Syntax: ( field: { \$lte: value } )
\$ne(not equal)	주어진 값과 일치하지 않는 값	Syntax: ( field: { \$ne: value } )
\$nin(not in)	주어진 배열 안에 속하지 않는 값	Syntax: ( field: { \$nin: [ value1, ... , valueN ] } )

## ■ 논리 연산자

Name	Example
\$or	Syntax: ( \$or: [ { expression1 }, ... , { expressionN } ] )
\$and	Syntax: ( \$and: [ { expression1 }, ... , { expressionN } ] )
\$not	Syntax: ( { field: { \$not: { operator-expression } } } )
\$nor	Syntax: ( { \$not: [ { expression1 }, ... , { expressionN } ] } )

## ■ 참고 사이트

- <https://docs.mongodb.com/manual/reference/operator/query/>