

2025년 상반기 K-디지털 트레이닝

# 자바스크립트 셀을 통한 MongoDB (기본)

[KB] IT's Your Life

- ✓ 데이터베이스를 tutorial로 선정하세요.

> use tutorial

- ✓ users 컬렉션에 username이 smith인 문서를 저장하세요.

> db.users.insert({username: "smith"})

- ✓ users 컬렉션에 username이 jones인 문서를 저장하세요.

> db.users.insert({username: " jones"})

- ✓ 앞에서 저장한 모든 문서를 출력하세요

> db.users.find()

- ✓ 앞에서 저장한 문서 중 하나 만 출력하세요

> db.users.findOne()

- ✓ **users 컬렉션에서 username이 "jones"인 문서를 찾아서 출력하세요.**

```
> db.users.find({username: "jones"})  
{ _id : ObjectId("4bf9bec90e32f82523389315"), username : "jones" }
```

- ✓ **users 컬렉션에서 username이 "jones" 또는 "smith"인 문서를 찾아서 출력하세요.**

```
> db.users.find({ $or: [  
  { username: "smith" },  
  { username: "jones" }  
]})
```

- ✓ **users 컬렉션에서 username이 smith인 문서에 country 키가 Canada 가 되도록 수정하세요.**

```
> db.users.update({username: "smith"}, {$set: {country: "Canada"}})
```

- ✓ **앞의 문서가 올바르게 수정되었는지 확인하세요.**

```
> db.users.find({username: "smith"})
{
  "_id" : ObjectId("4bf9ec440e32f82523389316"),
  "country" : "Canada",
  "username" : "smith"
}
```

- ✓ **users 컬렉션에서 username이 smith인 문서를 {country: "Canada"}로 대체하고 확인하세요.**

```
> db.users.update({username: "smith"}, {country: "Canada"})
> db.users.find({country: "Canada"})
{
  "_id" : ObjectId("4bf9ec440e32f82523389316"),
  "country" : "Canada"
}
```

- ✓ **users 컬렉션에서 country가 Canada인 문서를 {username: "smith", country: "Canada"}로 대체하고 확인하세요.**

```
> db.users.update({country: "Canada"}, {$set: {username: "smith", country: "Canada"}})
> db.users.find({username: "smith"})
```

- ✓ **users 컬렉션에서 username이 smith인 문서에서 country 키를 삭제하고, 삭제 여부를 확인하세요.**

```
> db.users.update({username: "smith"}, {$unset: {country: 1}})
```

```
> db.users.find({username: "smith"})
{
  "_id" : ObjectId("4bf9ec440e32f82523389316"),
  "username" : "smith"
}
```

- ✓ 데이터베이스 전체 목록을 출력하세요.

> show dbs

- ✓ 현재 사용 중인 데이터베이스의 컬렉션 목록을 출력하세요.

> show collections

- ✓ 현재 사용 중인 데이터베이스와 users 컬렉션의 상태를 출력하세요

> db.status()

> db.users.status()

- ✓ **users 컬렉션에서 username이 smith인 문서를 삭제하고, 삭제여부를 확인하세요.**

- > db.users.remove({username: "smith"})
  - > db.users.find({username: "smith"})

- ✓ **users 컬렉션의 모든 문서를 삭제하고, 삭제 여부를 확인하세요.**

- > db.users.remove({})
  - > db.users.find()

- ✓ **users 컬렉션을 삭제하세요.**

- > db.users.drop()



2025년 상반기 K-디지털 트레이닝

# 쿼리 작성하기 (기본)

---

[KB] IT's Your Life

- ✓ 다음 형태의 문서 20,000건을 추가하세요.
  - database: test
  - collection: product
  - 문서 필드
    - num: 순번 (0부터 시작)
    - name: '스마트폰 ' + 순번
  
- ✓ product 컬렉션의 전체 문서수를 출력하세요.

- ✓ 다음 형태의 문서 20,000건을 추가하세요.

```
use test;
for(let i=0; i<20000; i++) {
  db.product.insert({
    num: i,
    name: '스마트폰 ' + i
  });
}
```

- ✓ **product 컬렉션의 전체 문서수를 출력하세요.**

```
db.product.count();
```

- ✔ product 컬렉션의 문서를 num의 내림 차순으로 정렬하여 출력하세요.
  
- ✔ product 컬렉션의 문서를 num의 내림 차순으로 정렬하여 상위 10건을 출력하세요.
  
- ✔ product 컬렉션의 문서를 num의 내림 차순으로 정렬한 상태에서 다음을 처리하세요.
  - 한 페이지당 10건씩 출력
  - 6페이지에 해당하는 문서를 출력하세요.

- ✔ **product 컬렉션의 문서를 num의 내림 차순으로 정렬하여 출력하세요.**

```
db.product.find().sort({num: -1});
```

- ✔ **product 컬렉션의 문서를 num의 내림 차순으로 정렬하여 상위 10건을 출력하세요.**

```
db.product.find().sort({num: -1}).limit(10);
```

- ✔ **product 컬렉션의 문서를 num의 내림 차순으로 정렬한 상태에서 다음을 처리하세요.**

```
db.product.find()  
  .sort({num: -1})  
  .skip(10*(6-1))  
  .limit(10);
```

- ✔ **product 컬렉션에서 num이 15미만 이거나 19995 초과인 것을 출력하세요.**

```
db.product.find({ $or:
  [
    {num: {$lt: 15}},
    {num: {$gt: 19995}},
  ]
});
```

- ✔ **product 컬렉션에서 name이 '스마트폰 10', '스마트폰 100', '스마트폰 1000' 중에 하나이면 출력하세요.**

```
db.product.find({ name:
  {$in: ['스마트폰 10', '스마트폰 100', '스마트폰 1000']}
});
```

- ✓ product 컬렉션에서 num이 5보다 작은 문서를 출력하는데, name만 출력하세요.  
(\_id 출력하면 안됨)

```
db.product.find({num: {$lt: 5}}, {_id:0, name:1})
```