

```
test> show dbs
admin    40.00 KiB
config  60.00 KiB
lab01    72.00 KiB
local    40.00 KiB
mydb     40.00 KiB
myemp    96.00 KiB
test     40.00 KiB
test> use lab02
switched to db lab02
lab02> db
lab02
b/멋사/LikeLion-
Collab/curriculum/MyMongoDB/Lab02Workshop/setup.js)udDocs/CS_iCloud/comput
er.la
Uncaught:
SyntaxError: Unterminated string constant. (1:5)

> 1 | load("/Users/juanpark/Library/Mobile
Documents/com~apple~CloudDocs/CS_iCloud/computer.lab/멋사/LikeLion-
Collab/curriculum/MyMongoDB/Lab02Workshop/setup.js)
    |      ^
    2 |
lab02> load("/Users/juanpark/Library/Mobile
Documents/com~apple~Cl")dDocs/CS_iCloud/computer.lab/멋사/LikeLion-
Collab/curriculum/MyMongoDB/Lab02Workshop/setup.js
true
lab02> show collections
Product
lab02> db.Product.find()
[
  {
    _id: ObjectId('6849338e1098d5ed94ba6179'),
    Name: 'notebook',
    Price: 200,
    Category: 'material'
  },
  {
    _id: ObjectId('6849338e1098d5ed94ba617a'),
    Name: 'pencil',
    Price: 80,
    Category: 'material'
  },
  {
    _id: ObjectId('6849338e1098d5ed94ba617b'),
    Name: 'salad',
    Price: 220,
    Category: 'food'
  },
  {
    _id: ObjectId('6849338e1098d5ed94ba617c'),
    Name: 'others',
```

```

    Price: 20,
    Category: 'material'
  },
  {
    _id: ObjectId('6849338e1098d5ed94ba617d'),
    Name: 'bread',
    Price: 100,
    Category: 'food'
  }
]
lab02> db.Product.find({}, {_id:0})
[
  { Name: 'notebook', Price: 200, Category: 'material' },
  { Name: 'pencil', Price: 80, Category: 'material' },
  { Name: 'salad', Price: 220, Category: 'food' },
  { Name: 'others', Price: 20, Category: 'material' },
  { Name: 'bread', Price: 100, Category: 'food' }
]
lab02> print("1. 전체 출력")
1. 전체 출력

lab02> db.Product.find({}, {_id:0})
[
  { Name: 'notebook', Price: 200, Category: 'material' },
  { Name: 'pencil', Price: 80, Category: 'material' },
  { Name: 'salad', Price: 220, Category: 'food' },
  { Name: 'others', Price: 20, Category: 'material' },
  { Name: 'bread', Price: 100, Category: 'food' }
]
lab02> print("2.id 와 가격을 출력해 보자.")
2.id 와 가격을 출력해 보자.

lab02> db.Product.aggregate([{$project:{_id:1,Price:1}}]);
Uncaught:
SyntaxError: Unexpected token, expected "," (1:47)

> 1 | db.Product.aggregate([{$project:{_id:1,Price:1}}]);
    |                                     ^
    2 |

lab02> db.Product.aggregate([{$project:{_id:1,Price:1}}]);
[
  { _id: ObjectId('6849338e1098d5ed94ba6179'), Price: 200 },
  { _id: ObjectId('6849338e1098d5ed94ba617a'), Price: 80 },
  { _id: ObjectId('6849338e1098d5ed94ba617b'), Price: 220 },
  { _id: ObjectId('6849338e1098d5ed94ba617c'), Price: 20 },
  { _id: ObjectId('6849338e1098d5ed94ba617d'), Price: 100 }
]
lab02> print("가격만 출력 해보자.")
가격만 출력 해보자.

lab02> db.Product.aggregate([{$project:{Price:1}}]);
[
  { _id: ObjectId('6849338e1098d5ed94ba6179'), Price: 200 },

```

```
{ _id: ObjectId('6849338e1098d5ed94ba617a'), Price: 80 },
{ _id: ObjectId('6849338e1098d5ed94ba617b'), Price: 220 },
{ _id: ObjectId('6849338e1098d5ed94ba617c'), Price: 20 },
{ _id: ObjectId('6849338e1098d5ed94ba617d'), Price: 100 }
]
lab02> db.Product.aggregate([{$project:{_id:0, Price:1}}]);
[
  { Price: 200 },
  { Price: 80 },
  { Price: 220 },
  { Price: 20 },
  { Price: 100 }
]
lab02> print("3. Name 과 가격을 출력 해보자.")
3. Name 과 가격을 출력 해보자.

lab02> db.Product.aggregate([{$project:{Name:0, Price:1}}]);
MongoServerError[Location31253]: Invalid $project :: caused by :: Cannot
do inclusion on field Price in exclusion projection
lab02> db.Product.aggregate([{$project:{Name:1, Price:1}}]);
[
  {
    _id: ObjectId('6849338e1098d5ed94ba6179'),
    Name: 'notebook',
    Price: 200
  },
  {
    _id: ObjectId('6849338e1098d5ed94ba617a'),
    Name: 'pencil',
    Price: 80
  },
  {
    _id: ObjectId('6849338e1098d5ed94ba617b'),
    Name: 'salad',
    Price: 220
  },
  {
    _id: ObjectId('6849338e1098d5ed94ba617c'),
    Name: 'others',
    Price: 20
  },
  {
    _id: ObjectId('6849338e1098d5ed94ba617d'),
    Name: 'bread',
    Price: 100
  }
]
lab02> db.Product.aggregate([{$project:{_id:0,Name:1, Price:1}}]);
[
  { Name: 'notebook', Price: 200 },
  { Name: 'pencil', Price: 80 },
  { Name: 'salad', Price: 220 },
  { Name: 'others', Price: 20 },
  { Name: 'bread', Price: 100 }
```

```
]
lab02> print("4. 가격과 카테고리만 출력 해 보자. 단 가격별 오름차순으로 정렬을 한다.")
4. 가격과 카테고리만 출력 해 보자. 단 가격별 오름차순으로 정렬을 한다.

lab02> db.Product.aggregate([{$project:{_id:0,Price:1,Category:1}},{$sort:
{$Price:1}}]);
[
  { Price: 20, Category: 'material' },
  { Price: 80, Category: 'material' },
  { Price: 100, Category: 'food' },
  { Price: 200, Category: 'material' },
  { Price: 220, Category: 'food' }
]
lab02> db.Product.aggregate([{$project:{_id:0,Price:1,Category:1}},{$sort:
{$Price:-1}}]);
[
  { Price: 220, Category: 'food' },
  { Price: 200, Category: 'material' },
  { Price: 100, Category: 'food' },
  { Price: 80, Category: 'material' },
  { Price: 20, Category: 'material' }
]
lab02> print("5. 목록과 가격만을 출력하되 목록을 내림차순으로 정렬을 해보자.")
5. 목록과 가격만을 출력하되 목록을 내림차순으로 정렬을 해보자.

lab02> db.Product.aggregate([{$project:{_id:0,Name:1,Price:1}},{$sort:
{$Name:-1}}]);
[
  { Name: 'salad', Price: 220 },
  { Name: 'pencil', Price: 80 },
  { Name: 'others', Price: 20 },
  { Name: 'notebook', Price: 200 },
  { Name: 'bread', Price: 100 }
]
lab02> print("6. 이름만 출력하되 별칭을 주자")
6. 이름만 출력하되 별칭을 주자
lab02> db.Product.aggregate([{$group:{_id:"$Category",sum:
{$sum:"$Price"},avg:{$avg:"$Price"},count:{$sum:1}}}}]);
[
  { _id: 'material', sum: 300, avg: 100, count: 3 },
  { _id: 'food', sum: 320, avg: 160, count: 2 }
]
lab02> print("12.상품목록을 출력하고 그룹화 한 다음 개수를 구해보자")
12.상품목록을 출력하고 그룹화 한 다음 개수를 구해보자

lab02> edit
Opening an editor...

{ _id: "$Category", count: { $sum: "$count" } } } ] ); 1, count: {
$literal: 1 } } }, { $group:
[ { _id: 'material', count: 3 }, { _id: 'food', count: 2 } ]
lab02> edit
Opening an editor...
```

```
lab02> db.Product.aggregate([ { $group: { _id: "$Category", count: { $sum:
"$count" } } } ] );
[ { _id: 'material', count: 0 }, { _id: 'food', count: 0 } ]
lab02> db.Product.aggregate([ { $group: { _id: "$Category", count: {
$sum:1 } } } ] );
[ { _id: 'material', count: 3 }, { _id: 'food', count: 2 } ]
lab02> print("$match 활용해보자.")
$match 활용해보자.
```

```
lab02> print("13. Name 에서 bread 를 찾아 출력 하자.")
13. Name 에서 bread 를 찾아 출력 하자.
```

```
lab02> db.Product.aggregate([{$project:{$_id:0,Name:1}},{$match:
{Name:"bread"}}]);
```

Uncaught:

SyntaxError: Unterminated string constant. (1:61)

```
> 1 | db.Product.aggregate([{$project:{$_id:0,Name:1}},{$match:
{Name:"bread"}}]);
```

```
    |
    2 |
```

^

```
lab02> db.Product.aggregate([{$project:{$_id:0,Name:1}},{$match:
{Name:"bread"}}]);
```

MongoServerError[Location40324]: Unrecognized pipeline stage name: 'project'

```
lab02> db.Product.aggregate([{$project:{$_id:0,Name:1}},{$match:
{Name:"bread"}}]);
```

```
[ { Name: 'bread' } ]
```

```
lab02> db.Product.aggregate([{$project:{$_id:0}},{$match:{Name:"bread"}}]);
[ { Name: 'bread', Price: 100, Category: 'food' } ]
```

```
lab02> print("14. Category 에서 food 만 출력 해보자.")
```

14. Category 에서 food 만 출력 해보자.

```
lab02> db.Product.aggregate([{$match:{Category:"food"}}]);
```

```
[
```

```
  {
    _id: ObjectId('6849338e1098d5ed94ba617b'),
    Name: 'salad',
    Price: 220,
    Category: 'food'
  },
```

```
  {
    _id: ObjectId('6849338e1098d5ed94ba617d'),
    Name: 'bread',
    Price: 100,
    Category: 'food'
  }
]
```

```
]
```

```
lab02> edit(" ")
Opening an editor...
```

```
lab02> db.Product.aggregate([{$match: { Category: "food" } }]);
```

```
[
```

```

{
  _id: ObjectId('6849338e1098d5ed94ba617b'),
  Name: 'salad',
  Price: 220,
  Category: 'food'
},
{
  _id: ObjectId('6849338e1098d5ed94ba617d'),
  Name: 'bread',
  Price: 100,
  Category: 'food'
}
]
b/멋사/LikeLion-
Collab/curriculum/MyMongoDB/Lab02Workshop/test.js")udDocs/CS_iCloud/comput
er.la
lab02> db.Product.aggregate([ { $match: { Category: "food" } }, { $group:
{ _id: "$Category", max: { $max: "$Price" }, min: { $min: "$Price" }, sum:
{ $sum: "$Price" }, avg: { $avg: "$Price" }, count: { $sum: 1 } } } ] );
[ { _id: 'food', max: 220, min: 100, sum: 320, avg: 160, count: 2 } ]
lab02> exit
mini→ Lab02Workshop git:(main) x exit
mini→ Lab02Workshop git:(main)
2026;2$y2027;1$y2031;2$y2048;2$y0u62;22c11;rgb:2828/2c2c/3434
4
4
4
4
mini→ Lab02Workshop git:(main) script -q ./test.txt
mini→ Lab02Workshop git:(main) x mongosh
Current Mongosh Log ID: 6849441ed6bacb4b17eff6ec
Connecting to:          mongodb://127.0.0.1:27017/?
directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2
Using MongoDB:          8.0.10
Using Mongosh:           2.5.2

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

-----
The server generated these startup warnings when booting
2025-06-10T10:57:40.591+09:00: Access control is not enabled for the
database. Read and write access to data and configuration is unrestricted
-----

test> show dbs
admin      40.00 KiB
config    108.00 KiB
lab01      72.00 KiB
lab02      40.00 KiB
local      40.00 KiB
mydb       40.00 KiB
myemp     96.00 KiB
test       40.00 KiB
test> use lab02

```

```
switched to db lab02
lab02> db.Product.aggregate([ { $match: { Category: "food" } }, { $group:
{ _id: "$Category", max: { $max: "$Price" }, min: { $min: "$Price" }, sum:
{ $sum: "$Price" }, avg: { $avg: "$Price" }, count: { $sum: 1 } } } ] );
[ { _id: 'food', max: 220, min: 100, sum: 320, avg: 160, count: 2 } ]
lab02> load("/Users/juanpark/Library/Mobile
Documents/com~apple~CloudDocs/CS iCloud/computer.lab/멋사/LikeLion-
Collab/curriculum/MyMongoDB/Lab02Workshop/test.js")
true
lab02> db.Product.aggregate([ { $group: { _id: "$Category", count: {
$sum:1 } } } ] );
[ { _id: 'food', count: 2 }, { _id: 'material', count: 3 } ]
lab02> db.Product.aggregate([ { $project: { Category: 1, count: {
$literal: 1 } } }, { $group: { _id: "$Category", count: { $sum: "$count" }
} } ] );
[ { _id: 'material', count: 3 }, { _id: 'food', count: 2 } ]
lab02> db.Product.aggregate([{$group:{_id:"$Category",sum:
{$sum:"$Price"},avg:{$avg:"$Price"},count:{$sum:1}}}] );
[
  { _id: 'material', sum: 300, avg: 100, count: 3 },
  { _id: 'food', sum: 320, avg: 160, count: 2 }
]
lab02> exit
```