Go to c drive ->program files->mongodb->server->4.0->bin->erase the path and write CMD->enter

Practical 1- MongoDB basics

a)Write a mongoDB query to create and drop database.
Code:
Creating database:

- use tyit2023-24

Drop database

- db.dropDatabase()

b) Write a MongoDB query to create, display and drop collection.
Code:

Creating collection using explicit with capped collection with specific size

- db.createCollection(“Student”,{ capped:true, autoIndexId:true, size:641232, max:200 })

Displaying collection

- show collections

Drop collection

- db.Student.drop()

b) Write a MongoDB query to insert ,update and delete document.
Code:
Inserting document in collection

- db.Student1.insertOne({"Rollno":71, "Name":"Safwan", "Class":"TyBScIt", "Gender":"Male", "Age":20}

- db.Student1.insertMany([{"Rollno":71,"Name":"Safwan","Class":"TyBScIt","Gender":"Male","Age":20}])

Practical 2

```
> use College
switched to db College
> db.createCollection("student")
{ "ok" : 1 }
```

```
>
db.student.insert({"stu_id":1,"stu_name":"s1","stu_add":"nerul","phone_no":9587632458,"age":21})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":2,"stu_name":"s2","stu_add":"panvel","phone_no":9587632459,"age":22})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":3,"stu_name":"s3","stu_add":"panvel","phone_no":9587632451,"age":23})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":4,"stu_name":"s4","stu_add":"seawoods","phone_no":9587632452,"age":24})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":5,"stu_name":"s5","stu_add":"nerul","phone_no":9587632453,"age":25})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":6,"stu_name":"s6","stu_add":"panvel","phone_no":9587632454,"age":26})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":7,"stu_name":"s7","stu_add":"nerul","phone_no":9587632455,"age":27})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":8,"stu_name":"s8","stu_add":"nerul","phone_no":9587632456,"age":27})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":9,"stu_name":"s9","stu_add":"seawoods","phone_no":9587632457,"age":28})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":10,"stu_name":"s10","stu_add":"seawoods","phone_no":9587632458,"age":29})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":11,"stu_name":"s11","stu_add":"panvel","phone_no":9587632489,"age":30})
WriteResult({ "nInserted" : 1 })
```

```
>
db.student.insert({"stu_id":12,"stu_name":"s12","stu_add":"nerul","phone_no":9587642489,"age"
:30})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":13,"stu_name":"s13","stu_add":"nerul","phone_no":9587656489,"age"
:25})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":14,"stu_name":"s14","stu_add":"seawoods","phone_no":9587656459,
"age":29})
WriteResult({ "nInserted" : 1 })
>
db.student.insert({"stu_id":15,"stu_name":"s15","stu_add":"seawoods","phone_no":9547656459,
"age":24})
WriteResult({ "nInserted" : 1 })
> db.student.find()
{ "_id" : ObjectId("64afc93a6f9dded42bd6b19c"), "stu_id" : 1, "stu_name" : "s1", "stu_add" :
"nerul", "phone_no" : 9587632458, "age" : 21 }
{ "_id" : ObjectId("64afc94a6f9dded42bd6b19d"), "stu_id" : 2, "stu_name" : "s2", "stu_add" :
"panvel", "phone_no" : 9587632459, "age" : 22 }
{ "_id" : ObjectId("64afc94e6f9dded42bd6b19e"), "stu_id" : 3, "stu_name" : "s3", "stu_add" :
"panvel", "phone_no" : 9587632451, "age" : 23 }
{ "_id" : ObjectId("64afc9586f9dded42bd6b19f"), "stu_id" : 4, "stu_name" : "s4", "stu_add" :
"seawoods", "phone_no" : 9587632452, "age" : 24 }
{ "_id" : ObjectId("64afc95d6f9dded42bd6b1a0"), "stu_id" : 5, "stu_name" : "s5", "stu_add" :
"nerul", "phone_no" : 9587632453, "age" : 25 }
{ "_id" : ObjectId("64afc9636f9dded42bd6b1a1"), "stu_id" : 6, "stu_name" : "s6", "stu_add" :
"panvel", "phone_no" : 9587632454, "age" : 26 }
{ "_id" : ObjectId("64afc9646f9dded42bd6b1a2"), "stu_id" : 7, "stu_name" : "s7", "stu_add" :
"nerul", "phone_no" : 9587632455, "age" : 27 }
{ "_id" : ObjectId("64afc9656f9dded42bd6b1a3"), "stu_id" : 8, "stu_name" : "s8", "stu_add" :
"nerul", "phone_no" : 9587632456, "age" : 27 }
{ "_id" : ObjectId("64afc9666f9dded42bd6b1a4"), "stu_id" : 9, "stu_name" : "s9", "stu_add" :
"seawoods", "phone_no" : 9587632457, "age" : 28 }
{ "_id" : ObjectId("64afc9666f9dded42bd6b1a5"), "stu_id" : 10, "stu_name" : "s10", "stu_add" :
"seawoods", "phone_no" : 9587632458, "age" : 29 }
{ "_id" : ObjectId("64afc9676f9dded42bd6b1a6"), "stu_id" : 11, "stu_name" : "s11", "stu_add" :
"panvel", "phone_no" : 9587632489, "age" : 30 }
{ "_id" : ObjectId("64afc9686f9dded42bd6b1a7"), "stu_id" : 12, "stu_name" : "s12", "stu_add" :
"nerul", "phone_no" : 9587642489, "age" : 30 }
{ "_id" : ObjectId("64afc9686f9dded42bd6b1a8"), "stu_id" : 13, "stu_name" : "s13", "stu_add" :
"nerul", "phone_no" : 9587656489, "age" : 25 }
```

```
{ "_id" : ObjectId("64afc9696f9dded42bd6b1a9"), "stu_id" : 14, "stu_name" : "s14", "stu_add" :
"seawoods", "phone_no" : 9587656459, "age" : 29 }
{ "_id" : ObjectId("64afc9696f9dded42bd6b1aa"), "stu_id" : 15, "stu_name" : "s15", "stu_add" :
"seawoods", "phone_no" : 9547656459, "age" : 24 }
> db.student.find().limit(1)
{ "_id" : ObjectId("64afc93a6f9dded42bd6b19c"), "stu_id" : 1, "stu_name" : "s1", "stu_add" :
"nerul", "phone_no" : 9587632458, "age" : 21 }
> db.student.find({"stu_add":"seawoods"})
{ "_id" : ObjectId("64afc9586f9dded42bd6b19f"), "stu_id" : 4, "stu_name" : "s4", "stu_add" :
"seawoods", "phone_no" : 9587632452, "age" : 24 }
{ "_id" : ObjectId("64afc9666f9dded42bd6b1a4"), "stu_id" : 9, "stu_name" : "s9", "stu_add" :
"seawoods", "phone_no" : 9587632457, "age" : 28 }
{ "_id" : ObjectId("64afc9666f9dded42bd6b1a5"), "stu_id" : 10, "stu_name" : "s10", "stu_add" :
"seawoods", "phone_no" : 9587632458, "age" : 29 }
{ "_id" : ObjectId("64afc9696f9dded42bd6b1a9"), "stu_id" : 14, "stu_name" : "s14", "stu_add" :
"seawoods", "phone_no" : 9587656459, "age" : 29 }
{ "_id" : ObjectId("64afc9696f9dded42bd6b1aa"), "stu_id" : 15, "stu_name" : "s15", "stu_add" :
"seawoods", "phone_no" : 9547656459, "age" : 24 }
> db.student.find({$and:[{"stu_add":"nerul"},{"age":{"$gt":23}}]})
{ "_id" : ObjectId("64afc95d6f9dded42bd6b1a0"), "stu_id" : 5, "stu_name" : "s5", "stu_add" :
"nerul", "phone_no" : 9587632453, "age" : 25 }
{ "_id" : ObjectId("64afc9646f9dded42bd6b1a2"), "stu_id" : 7, "stu_name" : "s7", "stu_add" :
"nerul", "phone_no" : 9587632455, "age" : 27 }
{ "_id" : ObjectId("64afc9656f9dded42bd6b1a3"), "stu_id" : 8, "stu_name" : "s8", "stu_add" :
"nerul", "phone_no" : 9587632456, "age" : 27 }
{ "_id" : ObjectId("64afc9686f9dded42bd6b1a7"), "stu_id" : 12, "stu_name" : "s12", "stu_add" :
"nerul", "phone_no" : 9587642489, "age" : 30 }
{ "_id" : ObjectId("64afc9686f9dded42bd6b1a8"), "stu_id" : 13, "stu_name" : "s13", "stu_add" :
"nerul", "phone_no" : 9587656489, "age" : 25 }
> db.student.find({"$and":[{"$or":[{"stu_add":"nerul"},{"stu_add":"panvel"}]},{"age":{"$lt":26}}]})
{ "_id" : ObjectId("64afc93a6f9dded42bd6b19c"), "stu_id" : 1, "stu_name" : "s1", "stu_add" :
"nerul", "phone_no" : 9587632458, "age" : 21 }
{ "_id" : ObjectId("64afc94a6f9dded42bd6b19d"), "stu_id" : 2, "stu_name" : "s2", "stu_add" :
"panvel", "phone_no" : 9587632459, "age" : 22 }
{ "_id" : ObjectId("64afc94e6f9dded42bd6b19e"), "stu_id" : 3, "stu_name" : "s3", "stu_add" :
"panvel", "phone_no" : 9587632451, "age" : 23 }
{ "_id" : ObjectId("64afc95d6f9dded42bd6b1a0"), "stu_id" : 5, "stu_name" : "s5", "stu_add" :
"nerul", "phone_no" : 9587632453, "age" : 25 }
{ "_id" : ObjectId("64afc9686f9dded42bd6b1a8"), "stu_id" : 13, "stu_name" : "s13", "stu_add" :
"nerul", "phone_no" : 9587656489, "age" : 25 }
> db.student.update({"stu_add":"nerul"},{"$set":{"stu_add":"kharghar"}},{multi:true})
WriteResult({ "nMatched" : 6, "nUpserted" : 0, "nModified" : 6 })
> db.student.find()
```

```
{ "_id" : ObjectId("64afc93a6f9dded42bd6b19c"), "stu_id" : 1, "stu_name" : "s1", "stu_add" :
"kharghar", "phone_no" : 9587632458, "age" : 21 }
{ "_id" : ObjectId("64afc94a6f9dded42bd6b19d"), "stu_id" : 2, "stu_name" : "s2", "stu_add" :
"panvel", "phone_no" : 9587632459, "age" : 22 }
{ "_id" : ObjectId("64afc94e6f9dded42bd6b19e"), "stu_id" : 3, "stu_name" : "s3", "stu_add" :
"panvel", "phone_no" : 9587632451, "age" : 23 }
{ "_id" : ObjectId("64afc9586f9dded42bd6b19f"), "stu_id" : 4, "stu_name" : "s4", "stu_add" :
"seawoods", "phone_no" : 9587632452, "age" : 24 }
{ "_id" : ObjectId("64afc95d6f9dded42bd6b1a0"), "stu_id" : 5, "stu_name" : "s5", "stu_add" :
"kharghar", "phone_no" : 9587632453, "age" : 25 }
{ "_id" : ObjectId("64afc9636f9dded42bd6b1a1"), "stu_id" : 6, "stu_name" : "s6", "stu_add" :
"panvel", "phone_no" : 9587632454, "age" : 26 }
{ "_id" : ObjectId("64afc9646f9dded42bd6b1a2"), "stu_id" : 7, "stu_name" : "s7", "stu_add" :
"kharghar", "phone_no" : 9587632455, "age" : 27 }
{ "_id" : ObjectId("64afc9656f9dded42bd6b1a3"), "stu_id" : 8, "stu_name" : "s8", "stu_add" :
"kharghar", "phone_no" : 9587632456, "age" : 27 }
{ "_id" : ObjectId("64afc9666f9dded42bd6b1a4"), "stu_id" : 9, "stu_name" : "s9", "stu_add" :
"seawoods", "phone_no" : 9587632457, "age" : 28 }
{ "_id" : ObjectId("64afc9666f9dded42bd6b1a5"), "stu_id" : 10, "stu_name" : "s10", "stu_add" :
"seawoods", "phone_no" : 9587632458, "age" : 29 }
{ "_id" : ObjectId("64afc9676f9dded42bd6b1a6"), "stu_id" : 11, "stu_name" : "s11", "stu_add" :
"panvel", "phone_no" : 9587632489, "age" : 30 }
{ "_id" : ObjectId("64afc9686f9dded42bd6b1a7"), "stu_id" : 12, "stu_name" : "s12", "stu_add" :
"kharghar", "phone_no" : 9587642489, "age" : 30 }
{ "_id" : ObjectId("64afc9686f9dded42bd6b1a8"), "stu_id" : 13, "stu_name" : "s13", "stu_add" :
"kharghar", "phone_no" : 9587656489, "age" : 25 }
{ "_id" : ObjectId("64afc9696f9dded42bd6b1a9"), "stu_id" : 14, "stu_name" : "s14", "stu_add" :
"seawoods", "phone_no" : 9587656459, "age" : 29 }
{ "_id" : ObjectId("64afc9696f9dded42bd6b1aa"), "stu_id" : 15, "stu_name" : "s15", "stu_add" :
"seawoods", "phone_no" : 9547656459, "age" : 24 }
> db.student.update({},{"$set":{"hobby":"singing"}},{multi:true})
WriteResult({ "nMatched" : 15, "nUpserted" : 0, "nModified" : 15 })
> db.student.find()
{ "_id" : ObjectId("64afc93a6f9dded42bd6b19c"), "stu_id" : 1, "stu_name" : "s1", "stu_add" :
"kharghar", "phone_no" : 9587632458, "age" : 21, "hobby" : "singing" }
{ "_id" : ObjectId("64afc94a6f9dded42bd6b19d"), "stu_id" : 2, "stu_name" : "s2", "stu_add" :
"panvel", "phone_no" : 9587632459, "age" : 22, "hobby" : "singing" }
{ "_id" : ObjectId("64afc94e6f9dded42bd6b19e"), "stu_id" : 3, "stu_name" : "s3", "stu_add" :
"panvel", "phone_no" : 9587632451, "age" : 23, "hobby" : "singing" }
{ "_id" : ObjectId("64afc9586f9dded42bd6b19f"), "stu_id" : 4, "stu_name" : "s4", "stu_add" :
"seawoods", "phone_no" : 9587632452, "age" : 24, "hobby" : "singing" }
{ "_id" : ObjectId("64afc95d6f9dded42bd6b1a0"), "stu_id" : 5, "stu_name" : "s5", "stu_add" :
"kharghar", "phone_no" : 9587632453, "age" : 25, "hobby" : "singing" }
```

```
{ "_id" : ObjectId("64afc9636f9dded42bd6b1a1"), "stu_id" : 6, "stu_name" : "s6", "stu_add" :
"panvel", "phone_no" : 9587632454, "age" : 26, "hobby" : "singing" }
{ "_id" : ObjectId("64afc9646f9dded42bd6b1a2"), "stu_id" : 7, "stu_name" : "s7", "stu_add" :
"kharghar", "phone_no" : 9587632455, "age" : 27, "hobby" : "singing" }
{ "_id" : ObjectId("64afc9656f9dded42bd6b1a3"), "stu_id" : 8, "stu_name" : "s8", "stu_add" :
"kharghar", "phone_no" : 9587632456, "age" : 27, "hobby" : "singing" }
{ "_id" : ObjectId("64afc9666f9dded42bd6b1a4"), "stu_id" : 9, "stu_name" : "s9", "stu_add" :
"seawoods", "phone_no" : 9587632457, "age" : 28, "hobby" : "singing" }
{ "_id" : ObjectId("64afc9666f9dded42bd6b1a5"), "stu_id" : 10, "stu_name" : "s10", "stu_add" :
"seawoods", "phone_no" : 9587632458, "age" : 29, "hobby" : "singing" }
{ "_id" : ObjectId("64afc9676f9dded42bd6b1a6"), "stu_id" : 11, "stu_name" : "s11", "stu_add" :
"panvel", "phone_no" : 9587632489, "age" : 30, "hobby" : "singing" }
{ "_id" : ObjectId("64afc9686f9dded42bd6b1a7"), "stu_id" : 12, "stu_name" : "s12", "stu_add" :
"kharghar", "phone_no" : 9587642489, "age" : 30, "hobby" : "singing" }
{ "_id" : ObjectId("64afc9686f9dded42bd6b1a8"), "stu_id" : 13, "stu_name" : "s13", "stu_add" :
"kharghar", "phone_no" : 9587656489, "age" : 25, "hobby" : "singing" }
{ "_id" : ObjectId("64afc9696f9dded42bd6b1a9"), "stu_id" : 14, "stu_name" : "s14", "stu_add" :
"seawoods", "phone_no" : 9587656459, "age" : 29, "hobby" : "singing" }
{ "_id" : ObjectId("64afc9696f9dded42bd6b1aa"), "stu_id" : 15, "stu_name" : "s15", "stu_add" :
"seawoods", "phone_no" : 9547656459, "age" : 24, "hobby" : "singing" }
```

Practical 2 Restaurants

```
2)> db.restaurent.find({},{"restaurant_id":1,"name":1,"borough":1,"cuisine":1})

3)> db.restaurent.find({},{"restaurant_id":1,"name":1,"borough":1,"cuisine":1,"_id":0})

5)> db.restaurent.find({"borough":"Bronx"},{})

6)> db.restaurent.find({"borough":"Bronx"},{}).limit(5)

7)> db.restaurent.find({"borough":"Bronx"},{}).skip(5).limit(5)

8)> db.restaurent.find({"grades.score":{"$gt": 90}})

9)> db.restaurent.find({$and:[{"grades.score":{"$gt":80}},{"grades.score":{"$lt":100}}]})

"10"
> db.restaurent.find({"address.coord.0":{"$lt":-95.754168}})

"11"
>
db.restaurent.find({$and:[{"cuisine":{"$ne":"American"}},{"grades.score":{"$gt":70}},{"address.coo
rd.0":{"$lt":-65.754168}}]})
```

"12"

```
>db.restaurent.find({"cuisine":{"$ne":"American"},"grades.score":{"$gt":70},"address.coord.0":{"$l
t":-65.754168}})
```

"13"

```
>db.restaurent.find({$and:[{"cuisine":{"$ne":"American"}},{"grades.grade":"A"},{"borough":{"$ne":
"Brooklyn"}}]}).sort({"cuisine":-1})
```

"14"

```
> db.restaurent.find({"name":/^Wil/},{"restaurant_id":1},{"name":1},{"borough":1},{"cuisine":1})
```

"15"

```
> db.restaurent.find({"name":/ces$/},{"restaurant_id":1},{"name":1},{"borough":1},{"cuisine":1})
> db.restaurent.find()
```

"16"

```
> db.restaurent.find({"name":/.*Reg.*/},{"restaurant_id":1},{"name":1},{"borough":1},{"cuisine":1})
```

"17"

```
>db.restaurent.find({"borough":"Bronx","cuisine":{"$in":["American","Chinese"]}},{"name":1,"resta
urant_id":1,"_id":0,"borough":1,"cuisine":1})
```

"18"

```
> db.restaurent.find({$or:[{"borough":"Staten Island"},{"borough":"Queens"},{"borough":"Bronxor
Brooklyn"}]},{"restaurant_id":1,"name":1,"borough":1,"cuisine":1})
```

"19"

```
> db.restaurent.find({"borough":{"$nin":["Staten Island","Queens","Bronx",
"Brooklyn"]}},{"restaurant_id":1,"name":1,"borough":1,"cuisine":1})
```

"20"

```
>db.restaurent.find({"grades.score":{"$lte":10}},{"_id":0,"restaurant_id":1,"name":1,"borough":1,"
cuisine":1})
```

"21"

```
>db.restaurent.find({$or:[{"name":/^Wil/},{"$and":[{"cuisine":{"$ne":"American"}},{"cuisine":{"$ne":
"Chinees"}}]}]},{_id:0,name:1,restaurant_id:1,borough:1,cuisine:1})
```

"22"

```
>db.restaurent.find({"grades.date":ISODate("2014-08-11T00:00:00Z"),"grades.grade":"A","grade
s.score":11},{_id:0,name:1,resaturant_is:1,grades:1})
```

"23"

```
>db.restaurent.find({"grades.1.date":ISODate("2014-08-11T00:00:00Z"),"grades.1.grade":"A","gr
ades.1.score":9},{name:1,resaturant_is:1,grades:1})
```

"24"

```
>db.restaurent.find({"address.coord.1":{$gt:42,$lte:52}},{"name":1,"resaturant_id":1,"address":1}
)
```

"25"

```
> db.restaurent.find().sort({"name":1})
```

"27"

```
> db.restaurent.find().sort({"cuisine":1,"borough":-1})
```

"28"

```
> db.restaurent.find({"address.street":{$exists:true}})
```

"29"

```
> db.restaurent.find({"address.coord":{$type:1}})
```

"31"

```
> db.restaurent.find({"name":/.*mon.*/},{"name":1,"borough":1,"address.coord":1})
```

"32"

```
> db.restaurent.find({"name":/^Mad/},{"name":1,"borough":1,"address.coord":1})
```

Practical 3:Implementing Aggregation

a. Write a MongoDB query to use sum, avg, min and max expression.
Code:

```
> db.mycollection.insert({"title":"Practical MongoDB","description":"Nosql","by":["Shakuntala
Gupta","Navin
Sabharwal"],"url":"www.google.com","tags":["mongodb","nosql","replica"],"keyword":["mongoDB"
,"NoSQL","SQL"],"likes":130,"comments":{"username":"amit","message":"I liked this book very
much","date":"23 aug 2010","like":20}})
```

```
> db.mycollection.insert({"title":"ASP.NET","description":"This book is all about learning C#
language","by":"Max donalds","url":"www.gfg.com","tags":["C#
language","Inheritance"],"keyword":["C#","Webforms"],"likes":200,"comments":{"username":"jone
s","message":"I liked this book very much.I hope you will publish books like this more","date":"03
aug 2011","like":30}})
```

```
> db.mycollection.insert({"title":"Advanced Web Programming","description":"This book is all
about learning C# language and its concept","by":"Vipul
Sambre","url":"www.javatpoint.com","tags":["C#
language","Inheritance"],"keyword":["C#","Webforms"],"likes":256,"comments":{"username":"jack
","message":"I liked this book very much.I hope you will publish books like this more","date":"03
aug 2011","like":45}})

> db.mycollection.insert({"title":"Internet of Things","description":"This book provides info about
IOT Devices","by":"Sumatti jha","url":"www.google.com","tags":["IOT","Electrical
devices"],"keyword":["sensors","actuators","IOT","electronics"],"likes":5463,"comments":{"userna
me":"sumit","message":"I liked this book very much.I hope you will publish books like this
more","date":"31 jan 2011","like":56}})

> db.mycollection.insert({"title":"Internet of Things","description":"This book provides info about
IOT Devices","by":"Sumatti jha","url":"www.google.com","tags":["IOT","Electrical
devices"],"keyword":["sensors","actuators","IOT","electronics"],"likes":456,"comments":{"userna
me":"anish","message":"I liked this book very much.I hope you will publish books like this
more","date":"31 jan 2011","like":56}})

> db.mycollection.aggregate([{$group:{_id:"$by_username", num_likes:{$sum:"$likes"}}}])
{ "_id" : null, "num_likes" : 6505 }

> db.mycollection.aggregate([{$group:{_id:"$by_username", num_likes:{$avg:"$likes"}}}])
{ "_id" : null, "num_likes" : 1301 }

> db.mycollection.aggregate([{$group:{_id:"$by_username", num_likes:{$min:"$likes"}}}])
{ "_id" : null, "num_likes" : 130 }

> db.mycollection.aggregate([{$group:{_id:"$by_username", num_likes:{$max:"$likes"}}}])
{ "_id" : null, "num_likes" : 5463 }
```

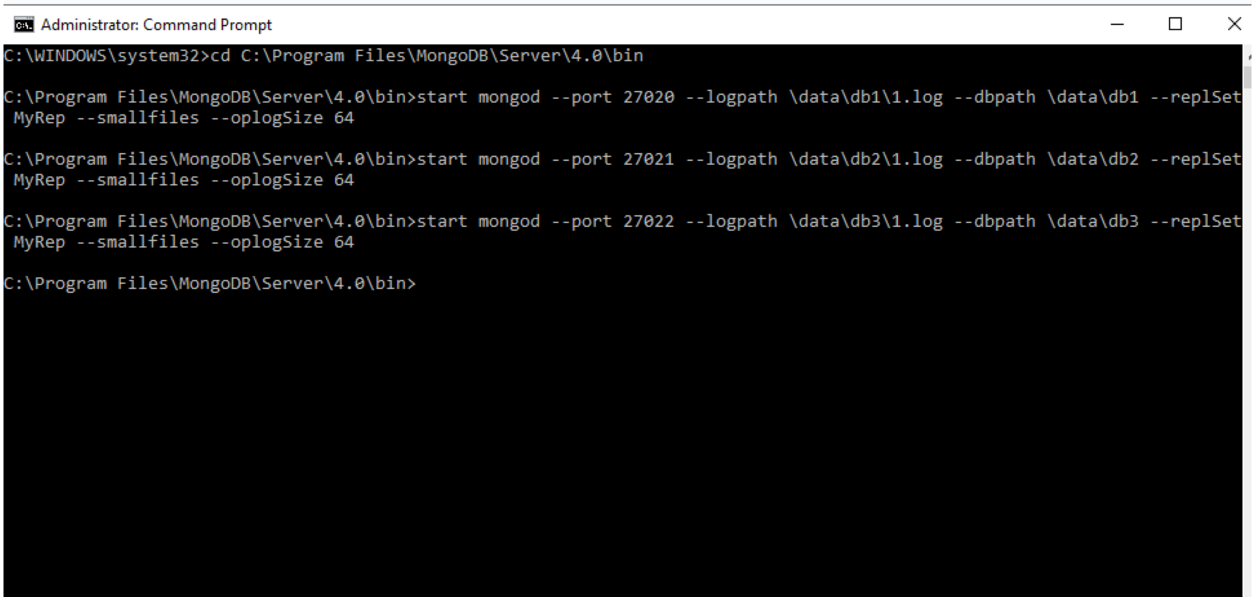
b. Write a MongoDB query to use push and addToSet expression.
Code:
Push: print all the values including duplicates
addToSet: removes duplicate values

```
> db.mycollection.aggregate([{$group:{_id:"$by_username", titles:{$push:"$title"}}}])
{ "_id" : null, "titles" : [ "Practical MongoDB", "ASP.NET", "Advanced Web Programming",
"Internet of Things", "Internet of Things" ] }

> db.mycollection.aggregate([{$group:{_id:"$by_username", titles:{$addToSet:"$title"}}}])
{ "_id" : null, "titles" : [ "Advanced Web Programming", "Internet of Things", "ASP.NET",
"Practical MongoDB" ] }
```

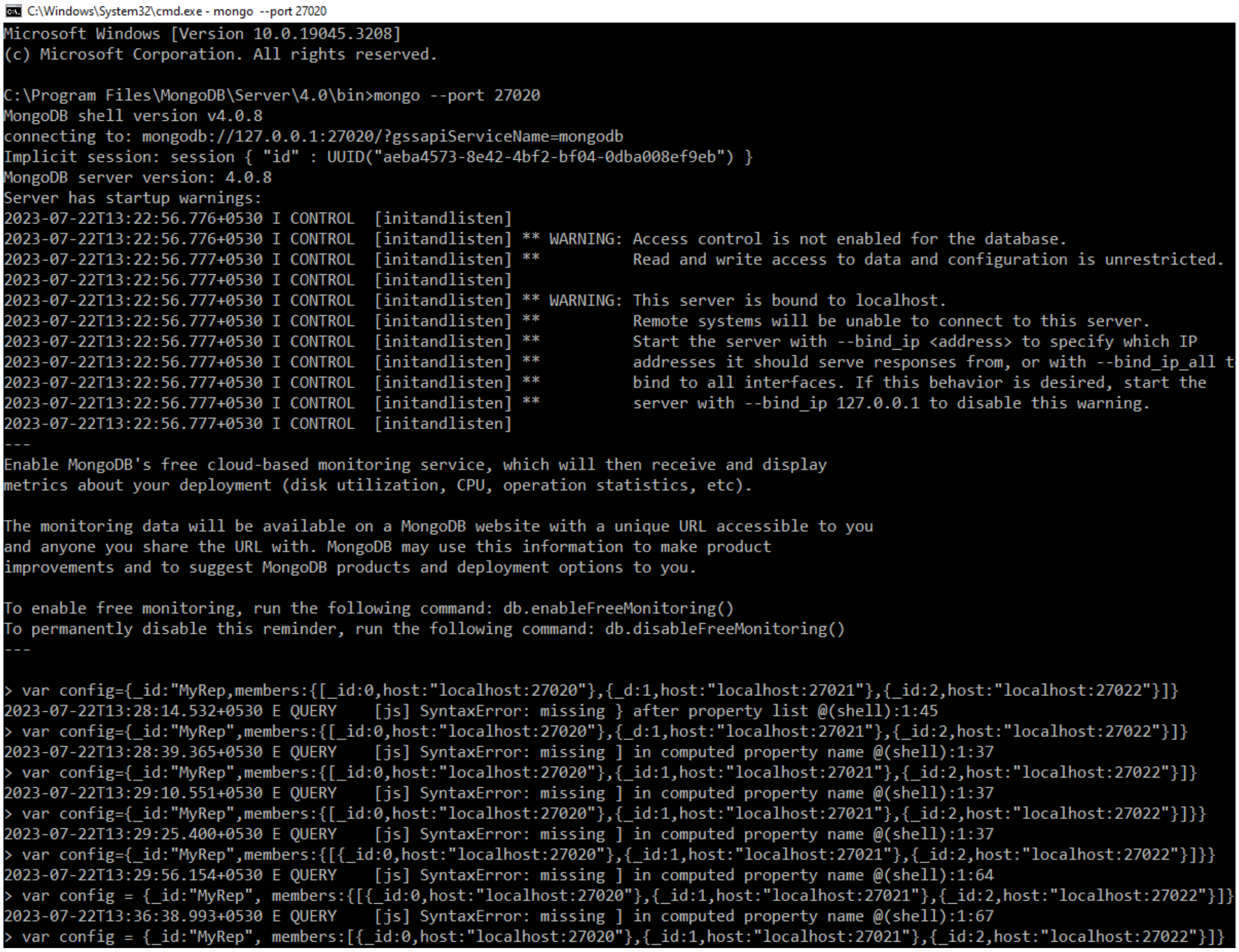
c. Write a MongoDB query to use first and last expression.
Code:
First: prints first title
Last: print last title

```
> db.mycollection.aggregate([{$group:{_id:"$by_username", titles:{$first:"$title"}}}])
{ "_id" : null, "titles" : "Practical MongoDB" }

> db.mycollection.aggregate([{$group:{_id:"$by_username", titles:{$last:"$title"}}}])
{ "_id" : null, "titles" : "Internet of Things" }
```

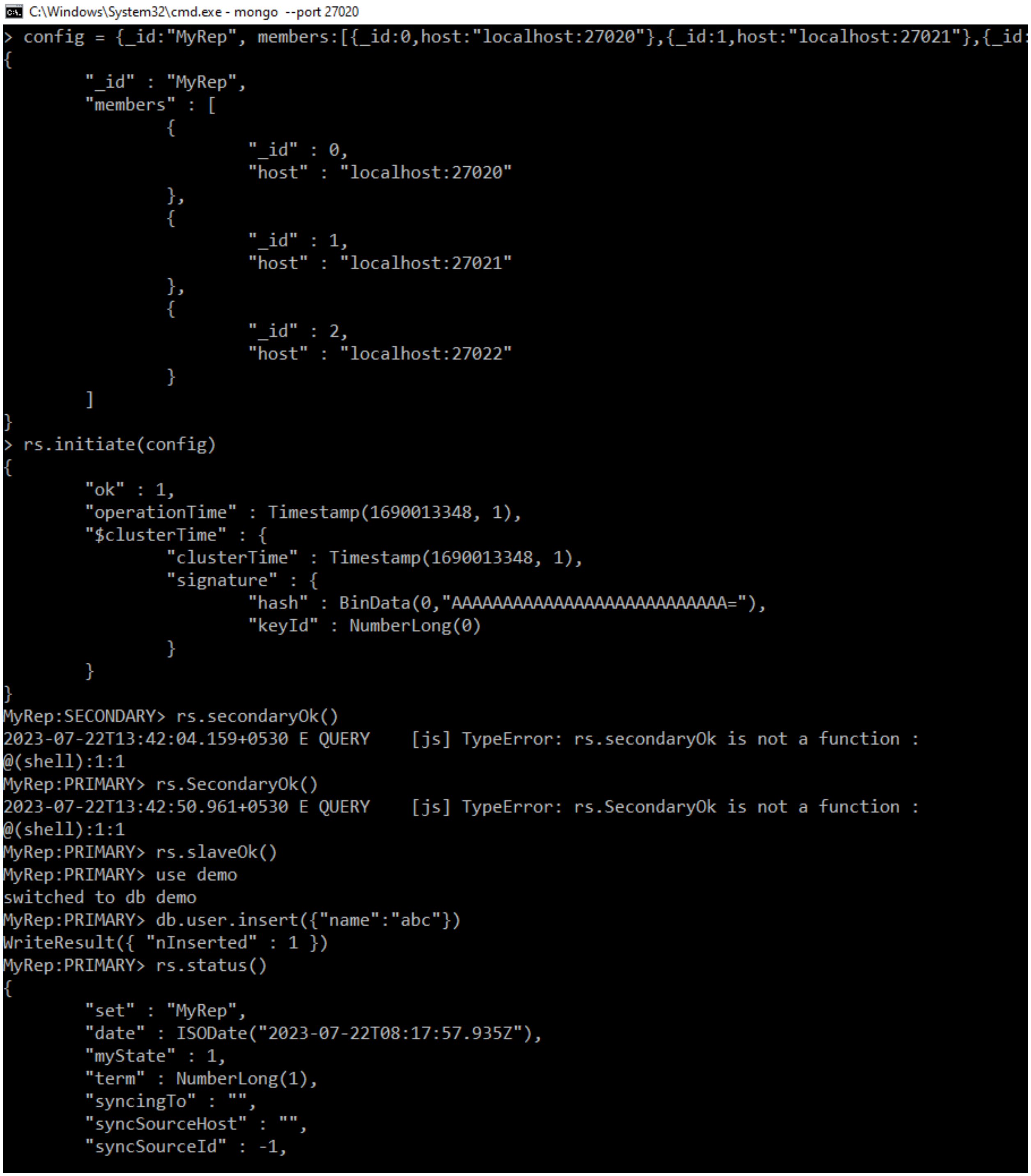
Practical 4: Replication, Backup and Restore



```
C:\WINDOWS\system32>cd C:\Program Files\MongoDB\Server\4.0\bin

C:\Program Files\MongoDB\Server\4.0\bin>start mongod --port 27020 --logpath \data\db1\1.log --dbpath \data\db1 --replSet
 MyRep --smallfiles --oplogSize 64

C:\Program Files\MongoDB\Server\4.0\bin>start mongod --port 27021 --logpath \data\db2\1.log --dbpath \data\db2 --replSet
 MyRep --smallfiles --oplogSize 64

C:\Program Files\MongoDB\Server\4.0\bin>start mongod --port 27022 --logpath \data\db3\1.log --dbpath \data\db3 --replSet
 MyRep --smallfiles --oplogSize 64

C:\Program Files\MongoDB\Server\4.0\bin>
```

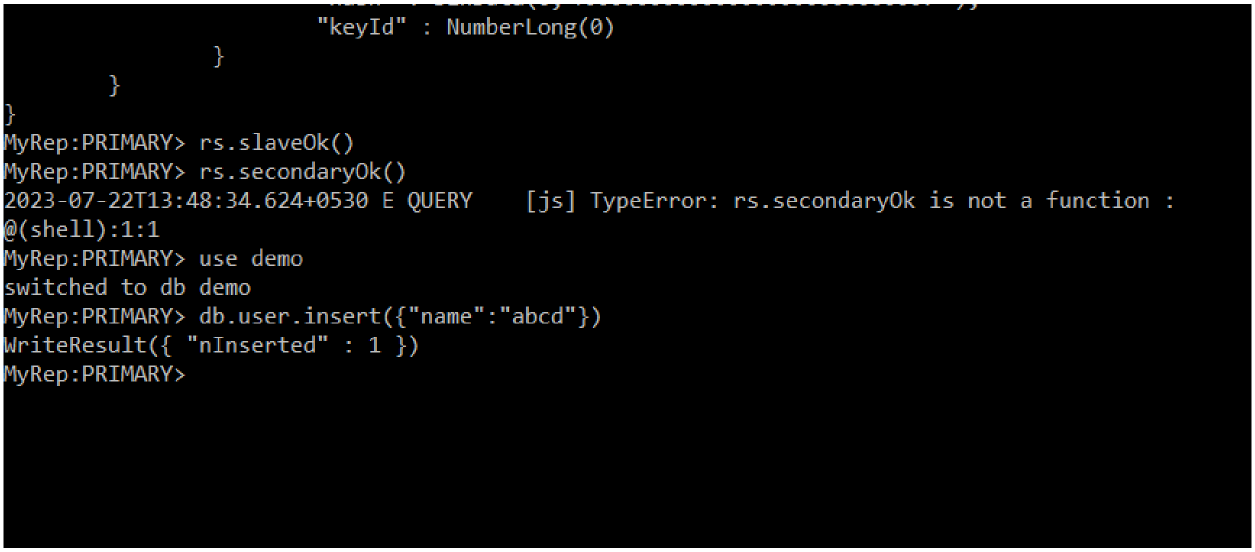


```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\MongoDB\Server\4.0\bin>mongo --port 27020
MongoDB shell version v4.0.8
connecting to: mongodb://127.0.0.1:27020/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("aeba4573-8e42-4bf2-bf04-0dba008ef9eb") }
MongoDB server version: 4.0.8
Server has startup warnings:
2023-07-22T13:22:56.776+0530 I CONTROL  [initandlisten]
2023-07-22T13:22:56.776+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2023-07-22T13:22:56.777+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2023-07-22T13:22:56.777+0530 I CONTROL  [initandlisten]
2023-07-22T13:22:56.777+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2023-07-22T13:22:56.777+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2023-07-22T13:22:56.777+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2023-07-22T13:22:56.777+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all t
2023-07-22T13:22:56.777+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2023-07-22T13:22:56.777+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2023-07-22T13:22:56.777+0530 I CONTROL  [initandlisten]
---
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()
---

> var config={_id:"MyRep,members:{[_id:0,host:"localhost:27020"},{_d:1,host:"localhost:27021"},{_id:2,host:"localhost:27022"}]}
2023-07-22T13:28:14.532+0530 E QUERY    [js] SyntaxError: missing } after property list @(shell):1:45
> var config={_id:"MyRep",members:{[_id:0,host:"localhost:27020"},{_d:1,host:"localhost:27021"},{_id:2,host:"localhost:27022"}]}
2023-07-22T13:28:39.365+0530 E QUERY    [js] SyntaxError: missing ] in computed property name @(shell):1:37
> var config={_id:"MyRep",members:{[_id:0,host:"localhost:27020"},{_id:1,host:"localhost:27021"},{_id:2,host:"localhost:27022"}]}
2023-07-22T13:29:10.551+0530 E QUERY    [js] SyntaxError: missing ] in computed property name @(shell):1:37
> var config={_id:"MyRep",members:{[_id:0,host:"localhost:27020"},{_id:1,host:"localhost:27021"},{_id:2,host:"localhost:27022"}]}}
2023-07-22T13:29:25.400+0530 E QUERY    [js] SyntaxError: missing ] in computed property name @(shell):1:37
> var config={_id:"MyRep",members:{[{_id:0,host:"localhost:27020"},{_id:1,host:"localhost:27021"},{_id:2,host:"localhost:27022"}]}}
2023-07-22T13:29:56.154+0530 E QUERY    [js] SyntaxError: missing ] in computed property name @(shell):1:64
> var config = {_id:"MyRep", members:{[{_id:0,host:"localhost:27020"},{_id:1,host:"localhost:27021"},{_id:2,host:"localhost:27022"}]}
2023-07-22T13:36:38.993+0530 E QUERY    [js] SyntaxError: missing ] in computed property name @(shell):1:67
> var config = {_id:"MyRep", members:[{_id:0,host:"localhost:27020"},{_id:1,host:"localhost:27021"},{_id:2,host:"localhost:27022"}]}
```

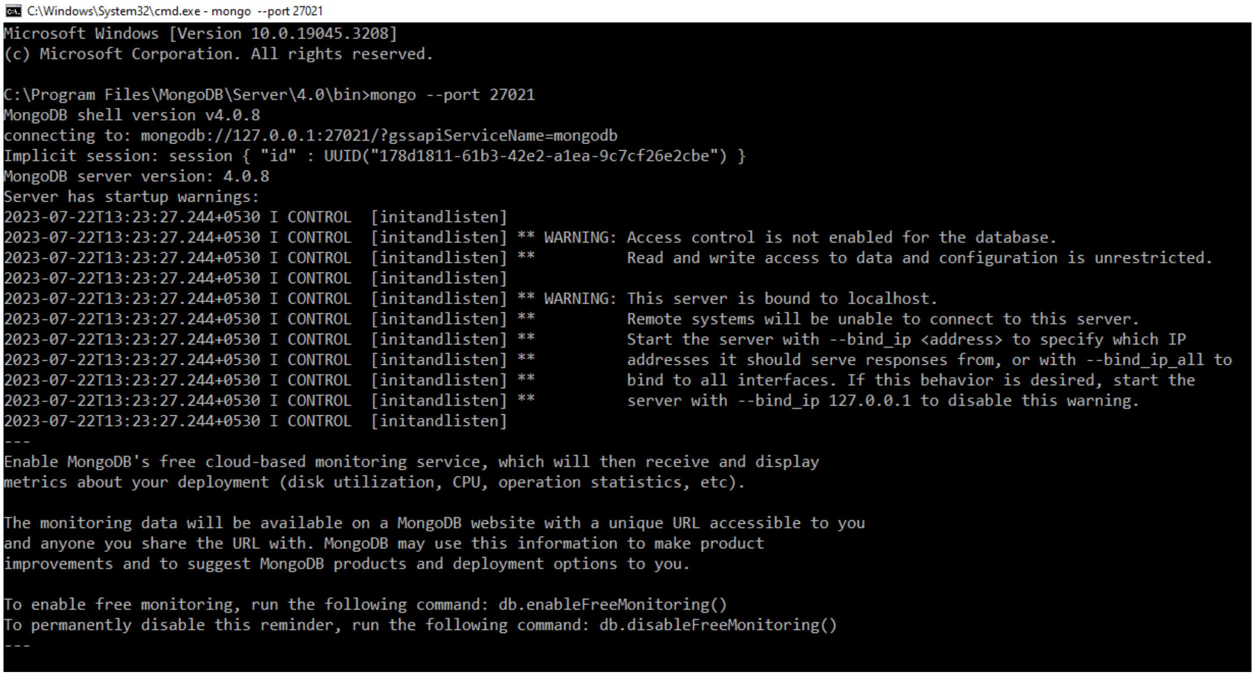


C:\Windows\System32\cmd.exe - mongo --port 27020

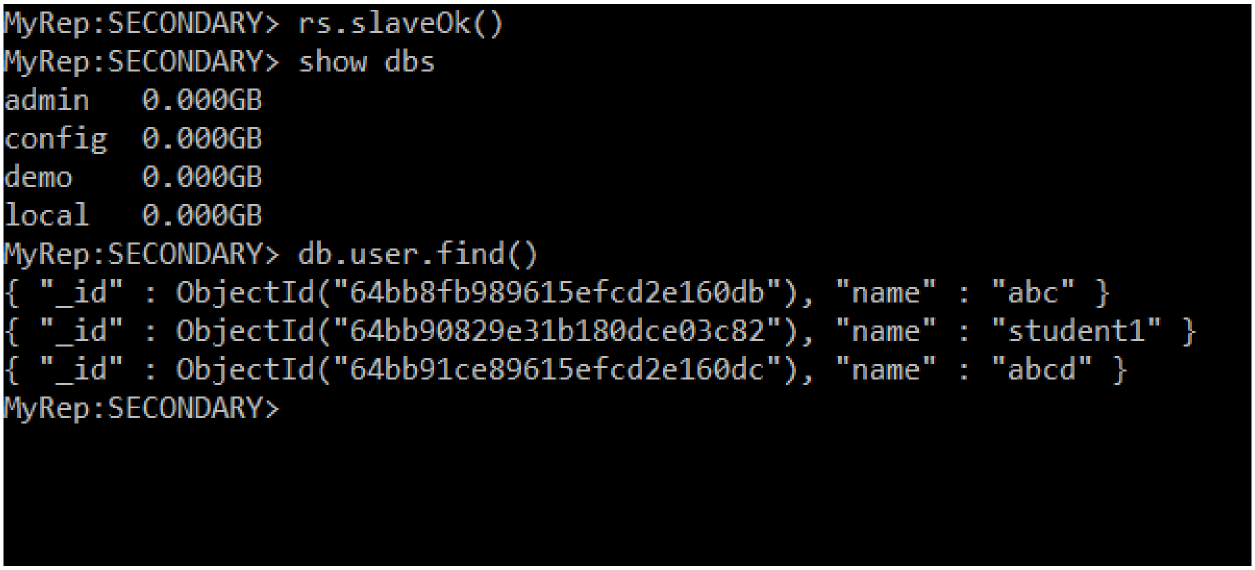
```
> config = {_id:"MyRep", members:[{_id:0,host:"localhost:27020"},{_id:1,host:"localhost:27021"},{_id:
{
        "_id" : "MyRep",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "localhost:27020"
                },
                {
                        "_id" : 1,
                        "host" : "localhost:27021"
                },
                {
                        "_id" : 2,
                        "host" : "localhost:27022"
                }
        ]
}
> rs.initiate(config)
{
        "ok" : 1,
        "operationTime" : Timestamp(1690013348, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1690013348, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
MyRep:SECONDARY> rs.secondaryOk()
2023-07-22T13:42:04.159+0530 E QUERY    [js] TypeError: rs.secondaryOk is not a function :
@(shell):1:1
MyRep:PRIMARY> rs.SecondaryOk()
2023-07-22T13:42:50.961+0530 E QUERY    [js] TypeError: rs.SecondaryOk is not a function :
@(shell):1:1
MyRep:PRIMARY> rs.slaveOk()
MyRep:PRIMARY> use demo
switched to db demo
MyRep:PRIMARY> db.user.insert({"name":"abc"})
WriteResult({ "nInserted" : 1 })
MyRep:PRIMARY> rs.status()
{
        "set" : "MyRep",
        "date" : ISODate("2023-07-22T08:17:57.935Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
```



```
                        "keyId" : NumberLong(0)
                }
        }
}
MyRep:PRIMARY> rs.slaveOk()
MyRep:PRIMARY> rs.secondaryOk()
2023-07-22T13:48:34.624+0530 E QUERY    [js] TypeError: rs.secondaryOk is not a function :
@(shell):1:1
MyRep:PRIMARY> use demo
switched to db demo
MyRep:PRIMARY> db.user.insert({"name":"abcd"})
WriteResult({ "nInserted" : 1 })
MyRep:PRIMARY>
```



```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\MongoDB\Server\4.0\bin>mongo --port 27021
MongoDB shell version v4.0.8
connecting to: mongodb://127.0.0.1:27021/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("178d1811-61b3-42e2-a1ea-9c7cf26e2cbe") }
MongoDB server version: 4.0.8
Server has startup warnings:
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten]
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten]
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2023-07-22T13:23:27.244+0530 I CONTROL  [initandlisten]
---
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()
---
```



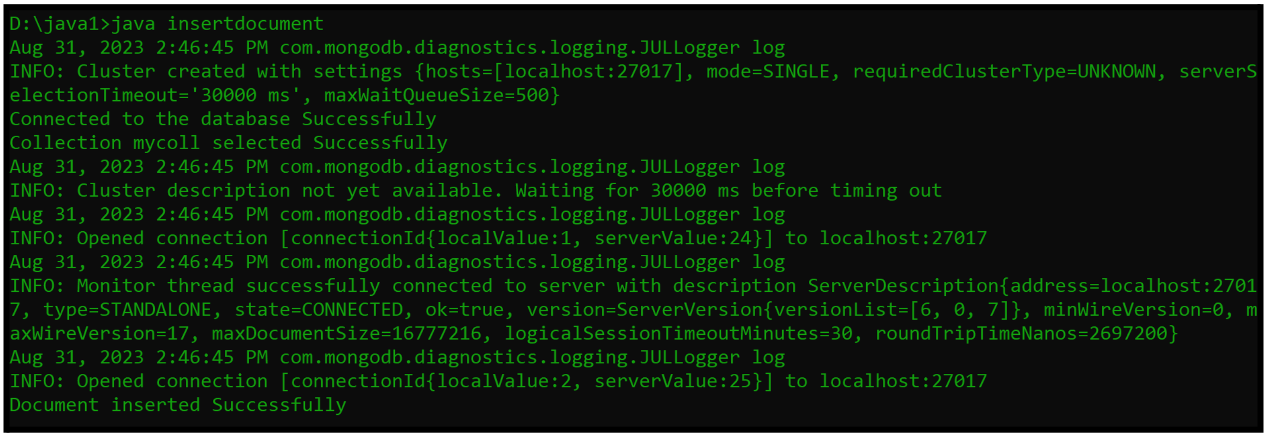
```
MyRep:SECONDARY> rs.slaveOk()
MyRep:SECONDARY> show dbs
admin   0.000GB
config  0.000GB
demo    0.000GB
local   0.000GB
MyRep:SECONDARY> db.user.find()
{ "_id" : ObjectId("64bb8fb989615efcd2e160db"), "name" : "abc" }
{ "_id" : ObjectId("64bb90829e31b180dce03c82"), "name" : "student1" }
{ "_id" : ObjectId("64bb91ce89615efcd2e160dc"), "name" : "abcd" }
MyRep:SECONDARY>
```

Practical 5:Java and MongoDB

a) Connecting Java with MongoDB and inserting, retrieving, updating and deleting.

Code:

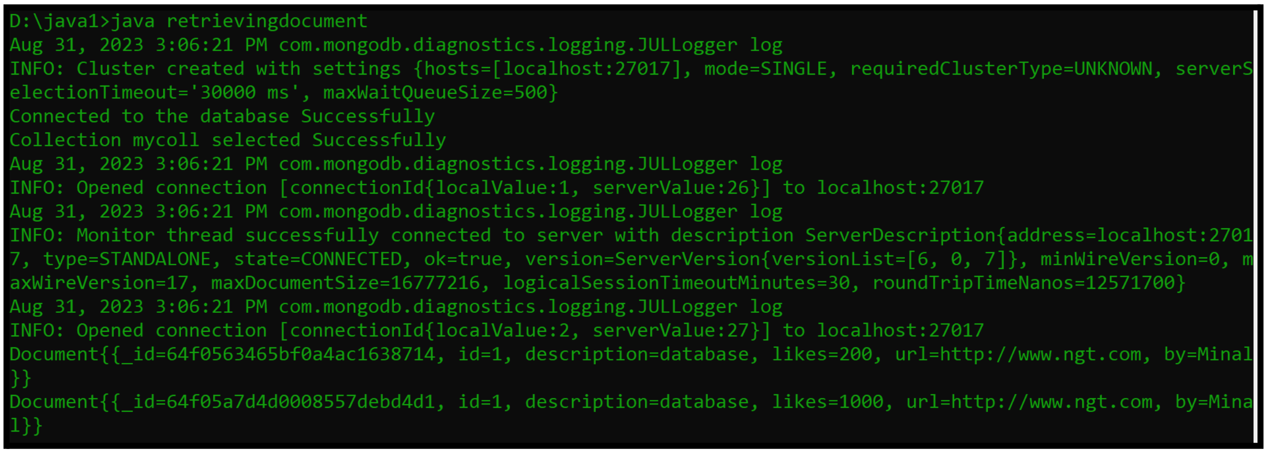
```
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;

import com.mongodb.MongoClient;
import com.mongodb.MongoCredential;

import org.bson.Document;

public class insertdocument
{
        public static void main(String args[])
        {
                MongoClient mongo=new MongoClient("localhost",27017);
                System.out.println("Connected to the database Successfully");

                MongoDatabase databse=mongo.getDatabase("mydb2");

                MongoCollection<Document> collection=databse.getCollection("mycoll");
                System.out.println("Collection mycoll selected Successfully");

                Document doc=new Document();
                doc.append("id",1);
                doc.append("description","database");
                doc.append("description","database");
                doc.append("likes",1000);
                doc.append("url","http://www.ngt.com");
                doc.append("by","Minal");

                collection.insertOne(doc);

                System.out.println("Document inserted Successfully");
        }
}
```



```
D:\java1>java insertdocument
Aug 31, 2023 2:46:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverS
electionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to the database Successfully
Collection mycoll selected Successfully
Aug 31, 2023 2:46:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Aug 31, 2023 2:46:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:24}] to localhost:27017
Aug 31, 2023 2:46:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:2701
7, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[6, 0, 7]}, minWireVersion=0, m
axWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=2697200}
Aug 31, 2023 2:46:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:25}] to localhost:27017
Document inserted Successfully
```

<u>Code:</u>

```
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;

import com.mongodb.MongoClient;
import com.mongodb.MongoCredential;

import org.bson.Document;
import java.util.Iterator;
import com.mongodb.client.FindIterable;

public class retrievingdocument
{
        public static void main(String args[])
        {
                MongoClient mongo=new MongoClient("localhost",27017);
                System.out.println("Connected to the database Successfully");

                MongoDatabase databse=mongo.getDatabase("mydb2");

                MongoCollection<Document> collection=databse.getCollection("mycoll");
                System.out.println("Collection mycoll selected Successfully");

                FindIterable<Document> iterDoc = collection.find();
                int i=1;

                Iterator it= iterDoc.iterator();

                while(it.hasNext())
                {
                    System.out.println(it.next());
                    i++;
                }
        }
}
```

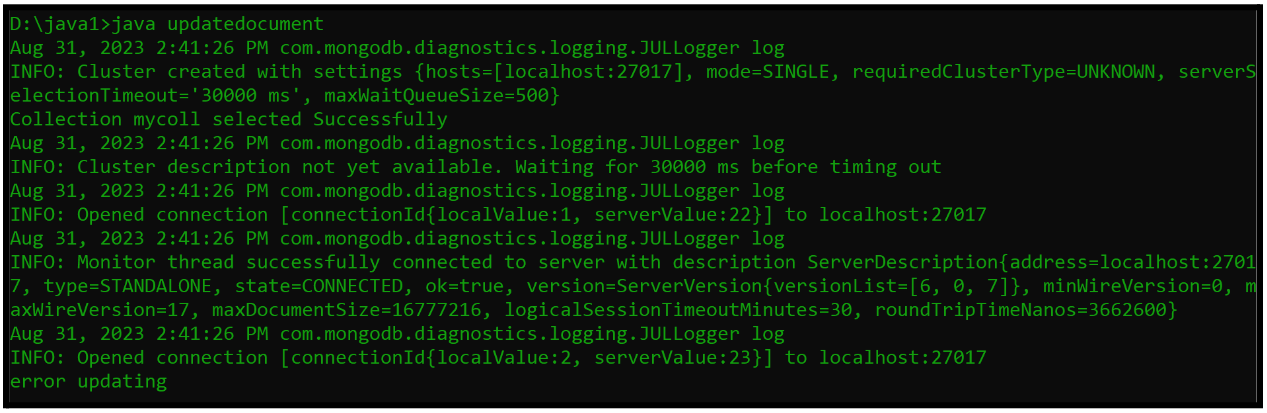


```
D:\java1>java retrievingdocument
Aug 31, 2023 3:06:21 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverS
electionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to the database Successfully
Collection mycoll selected Successfully
Aug 31, 2023 3:06:21 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:26}] to localhost:27017
Aug 31, 2023 3:06:21 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:2701
7, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[6, 0, 7]}, minWireVersion=0, m
axWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=12571700}
Aug 31, 2023 3:06:21 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:27}] to localhost:27017
Document{{_id=64f0563465bf0a4ac1638714, id=1, description=database, likes=200, url=http://www.ngt.com, by=Minal
}}
Document{{_id=64f05a7d4d0008557debd4d1, id=1, description=database, likes=1000, url=http://www.ngt.com, by=Mina
l}}
```

<u>Code:</u>

```
import java.net.UnknownHostException;
import com.mongodb.BasicDBObject;
import com.mongodb.DB;

import com.mongodb.DBCollection;
import com.mongodb.DBObject;

import com.mongodb.MongoClient;
import com.mongodb.WriteResult;

public class updatedocument
{
    public static void main(String[] args) throws UnknownHostException
    {
        MongoClient mongo=new MongoClient("localhost",27017);
```

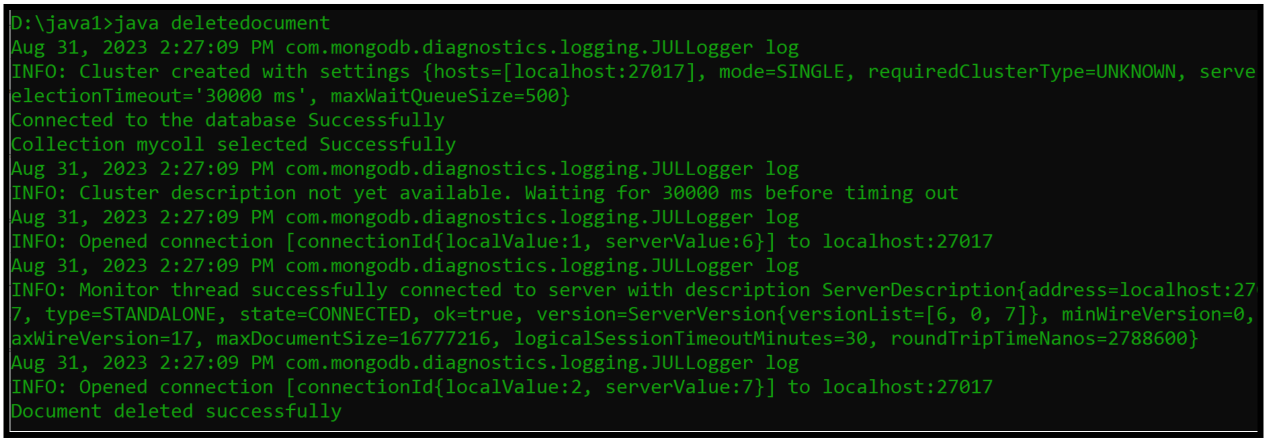
```
                DB db=mongo.getDB("mydb2");

                DBCollection col=db.getCollection("mycoll");
                System.out.println("Collection mycoll selected Successfully");

                DBObject query= new BasicDBObject("id",1);
                DBObject update= new BasicDBObject();

                update.put("$set", new BasicDBObject("likes",200));
                WriteResult result= col.update(query, update);
                System.out.println("error updating");


        }
}
```



```
D:\java1>java updatedocument
Aug 31, 2023 2:41:26 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverS
electionTimeout='30000 ms', maxWaitQueueSize=500}
Collection mycoll selected Successfully
Aug 31, 2023 2:41:26 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Aug 31, 2023 2:41:26 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:22}] to localhost:27017
Aug 31, 2023 2:41:26 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:2701
7, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[6, 0, 7]}, minWireVersion=0, m
axWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=3662600}
Aug 31, 2023 2:41:26 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:23}] to localhost:27017
error updating
```

Code:

```
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;

import com.mongodb.MongoClient;
import com.mongodb.MongoCredential;

import org.bson.Document;
import com.mongodb.client.model.Filters;//it stores filter property

public class deletedocument
{
        public static void main(String[]args)
        {
                MongoClient mongo=new MongoClient("localhost",27017);
                System.out.println("Connected to the database Successfully");
                MongoDatabase databse=mongo.getDatabase("mydb2");

                MongoCollection<Document> collection=databse.getCollection("mycoll");
                System.out.println("Collection mycoll selected Successfully");

                //delete my documents
                collection.deleteOne(Filters.eq("id",1));
                System.out.println("Document deleted successfully");
        }
}
```



```
D:\java1>java deletedocument
Aug 31, 2023 2:27:09 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serve
electionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to the database Successfully
Collection mycoll selected Successfully
Aug 31, 2023 2:27:09 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Aug 31, 2023 2:27:09 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:6}] to localhost:27017
Aug 31, 2023 2:27:09 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27
7, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[6, 0, 7]}, minWireVersion=0,
axWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=2788600}
Aug 31, 2023 2:27:09 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:7}] to localhost:27017
Document deleted successfully
```

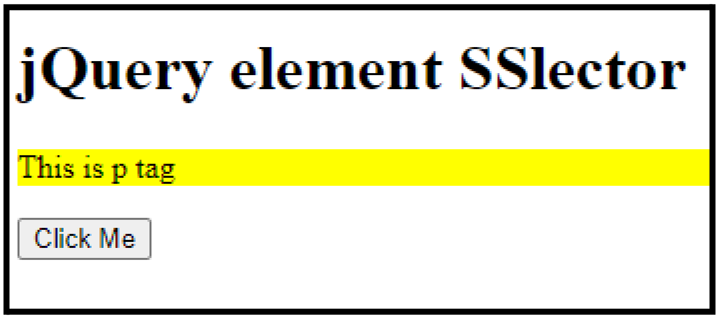
JQuery works on selection:
3 types of specifying the selection by use of :-
Name of the Elements
Id attribute
Class attribute

Q1)write a jquery to change the text content of the element on the button click.

<u>Code:</u>

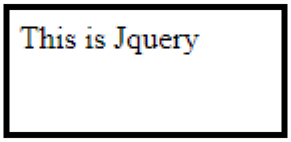
```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
    $("p").css("background-color", "yellow");
     $("button").click(function(){
        document.write("This is Jquery");
     });
  });
</script>
</head>
<body>
  <h1>jQuery element Selector</h1>
  <p>This is p tag</p>
  <button>Click Me</button>
</body>
</html>
```

<u>Output:</u>



jQuery element SSlector

This is p tag



This is Jquery

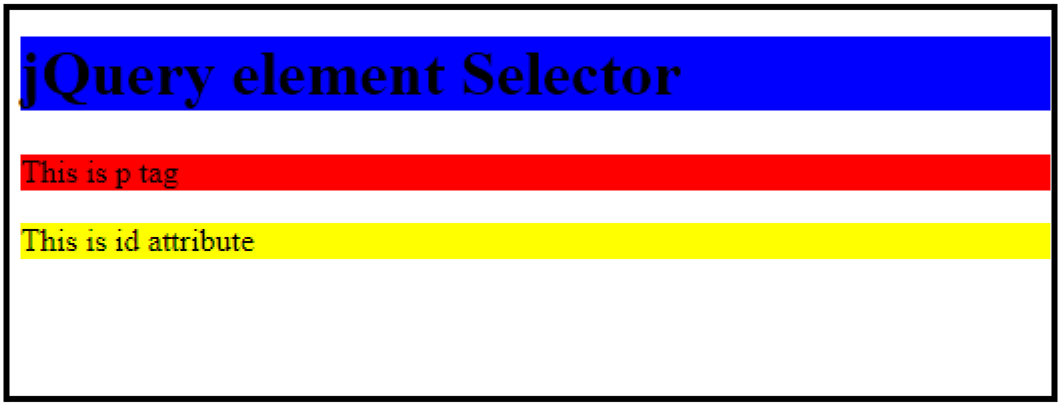
Q2)write a jquery to select elements by class name,id and element name.
2 paragraph tag and 1 h1 tag
In this example

Code:I

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
     $(".p1").css("background","red")
      $("#p2").css("background","yellow")
       $("h1").css("background","blue")
     });
</script>
</head>
<body>
  <h1>jQuery element Selector</h1>
  <p class="p1">This is p tag</p>
```

```
   <p id="p2">This is id attribute</p>
</body>
</html>
```

Output:



jQuery element Selector

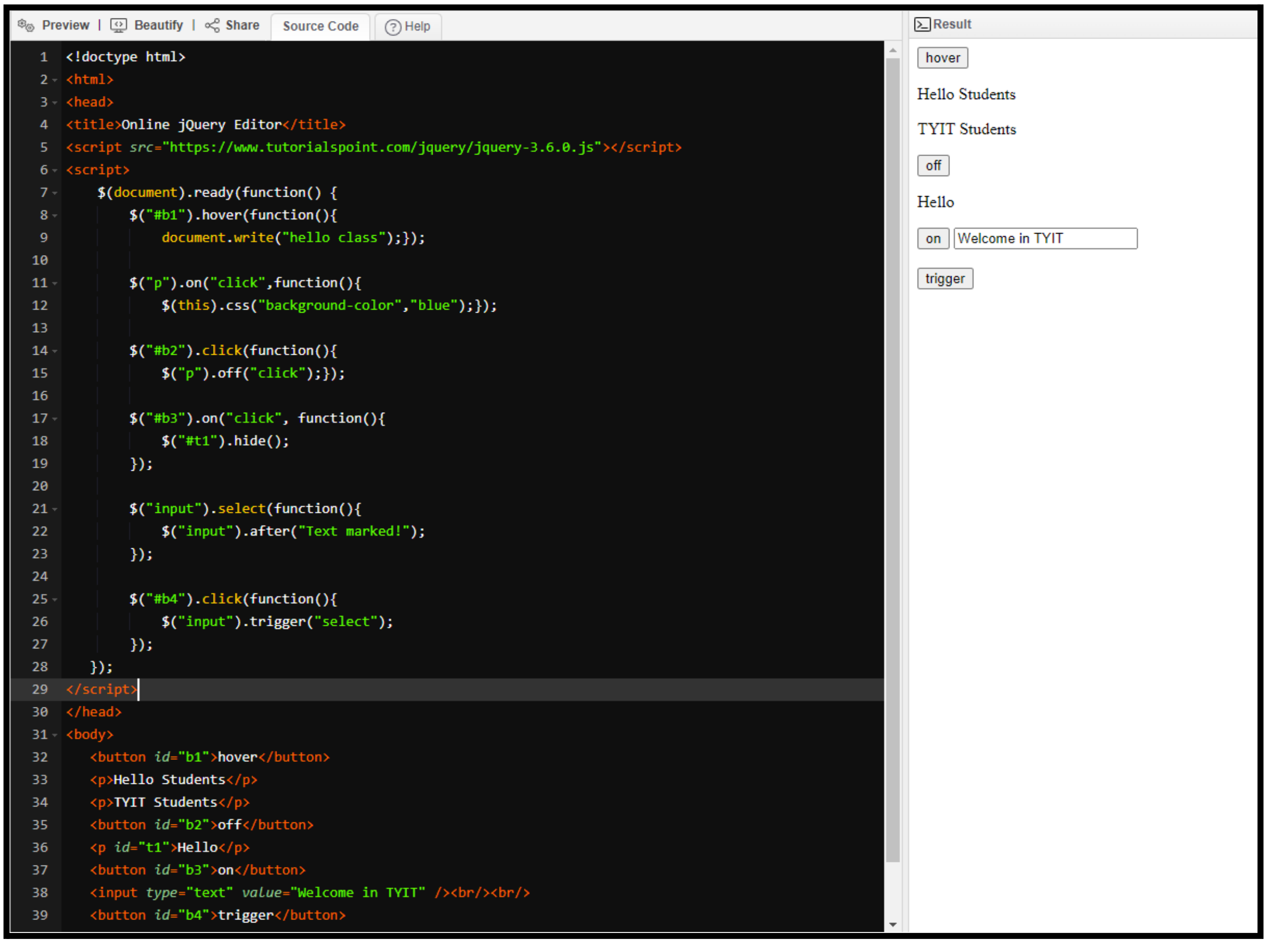
This is p tag

This is id attribute

Q3)Write a jquery to show the use of click(), hover(), on(), trigger() events.
Place four buttons as b1,b2,b3,b4
Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        $("#b1").hover(function(){
            document.write("hello class");});

        $("p").on("click",function(){
            $(this).css("background-color","blue");});

        $("#b2").click(function(){
            $("p").off("click");});

        $("#b3").on("click", function(){
            $("#t1").hide();
        });

        $("input").select(function(){
            $("input").after("Text marked!");
        });

        $("#b4").click(function(){
```

```
        $("input").trigger("select");
      });
  });
</script>
</head>

<body>
  <button id="b1">hover</button>
  <p>Hello Students</p>
  <p>TYIT Students</p>
  <button id="b2">off</button>
  <p id="t1">Hello</p>
  <button id="b3">on</button>
  <input type="text" value="Welcome in TYIT" /><br/><br/>
  <button id="b4">trigger</button>
</body>
</html>
```



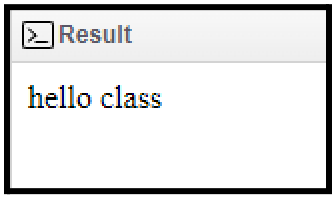
```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        $("#b1").hover(function(){
            document.write("hello class");});

        $("p").on("click",function(){
            $(this).css("background-color","blue");});

        $("#b2").click(function(){
            $("p").off("click");});

        $("#b3").on("click", function(){
            $("#t1").hide();
        });

        $("input").select(function(){
            $("input").after("Text marked!");
        });

        $("#b4").click(function(){
            $("input").trigger("select");
        });
    });
</script>
</head>
<body>
   <button id="b1">hover</button>
   <p>Hello Students</p>
   <p>TYIT Students</p>
   <button id="b2">off</button>
   <p id="t1">Hello</p>
   <button id="b3">on</button>
   <input type="text" value="Welcome in TYIT" /><br/><br/>
   <button id="b4">trigger</button>
```

```
hover

Hello Students

TYIT Students

off

Hello

on  Welcome in TYIT

trigger
```

Output:

BEFORE

AFTER

Hover output: 

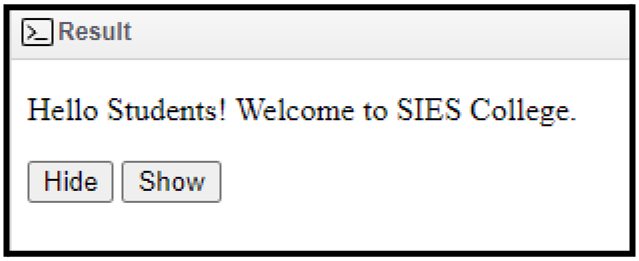
hello class

8.a) write a jquery to create animated show, hide effects.

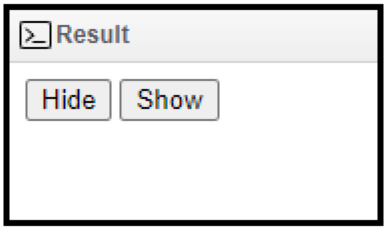
Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
   $(document).ready(function() {
      $("#b1").click(function(){
         $("p").hide();
      });

      $("#b2").click(function(){
         $("p").show();
      });
  });
</script>
</head>
<body>
  <p>Hello Students! Welcome to SIES College.</p>
  <button id="b1">Hide</button>
  <button id="b2">Show</button>

</body>
</html>
```

Output:



Hello Students! Welcome to SIES College.

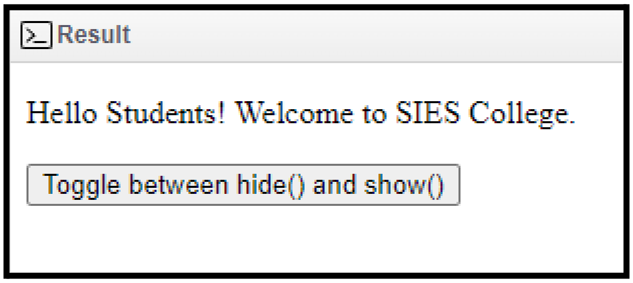


8.b) Write a jquery to create a simple toggle effect.

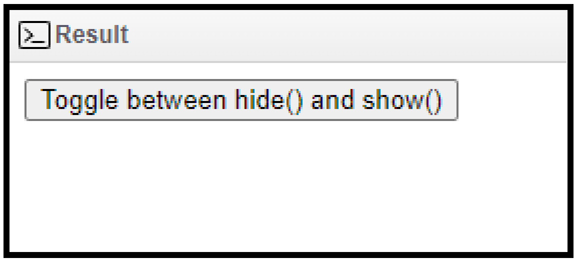
Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
   $(document).ready(function() {
      $("#b1").click(function(){
         $("p").toggle();
      });
  });
</script>
</head>
<body>
  <p>Hello Students! Welcome to SIES College.</p>
  <button id="b1">Toggle between hide() and show()</button>

</body>
</html>
```

Output:



Hello Students! Welcome to SIES College.



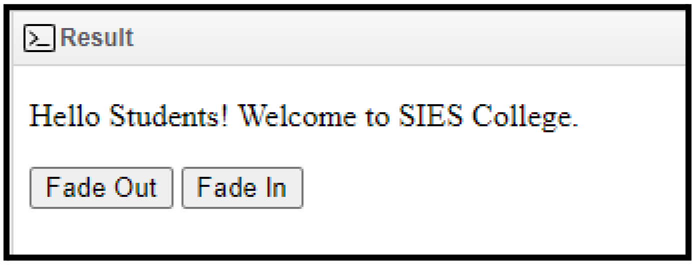
Q) Write a jquery to create a fadeOut(), fadeIn() effect.

Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        $("#b1").click(function(){
            $("p").fadeOut();
        });

        $("#b2").click(function(){
            $("p").fadeIn();
        });
  });
</script>
</head>
<body>
  <p>Hello Students! Welcome to SIES College.</p>
  <button id="b1">Fade Out</button>
  <button id="b2">Fade In</button>
</body>
</html>
```

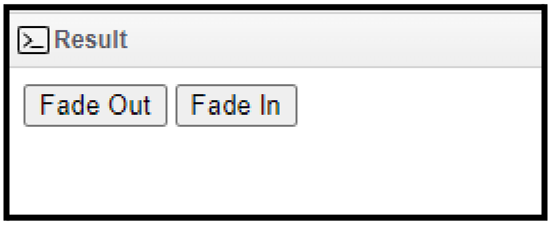
Output:

Fade In effect



Hello Students! Welcome to SIES College.

Fade Out effect

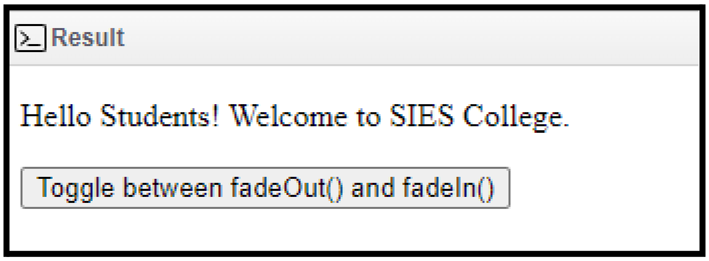


8.c) fadeToggle() function

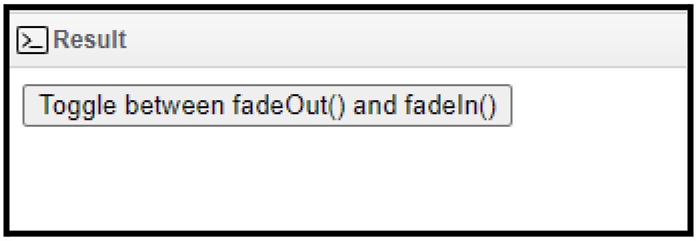
Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        $("#b1").click(function(){
            $("p").fadeToggle();
        });
   });
</script>
</head>
<body>
   <p>Hello Students! Welcome to SIES College.</p>
   <button id="b1">Toggle between fadeOut() and fadeIn()</button>
</body>
</html>
```

Outpu:



Hello Students! Welcome to SIES College.



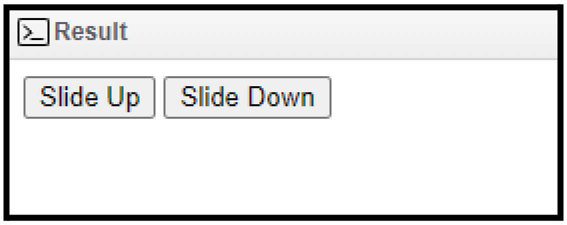
8.c)jQuery fading effects, jQuery Sliding effects

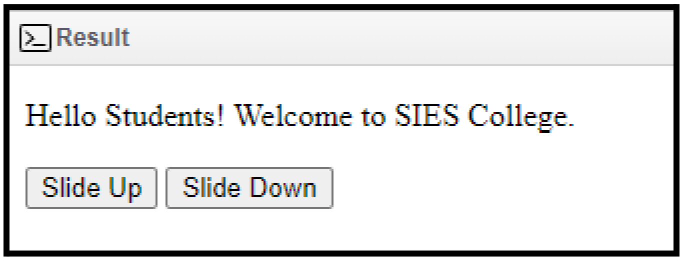
Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
```

```
<script>
    $(document).ready(function() {
        $("#b1").click(function(){
            $("p").slideUp();
        });

        $("#b2").click(function(){
            $("p").slideDown();
        });
   });
</script>
</head>
<body>
   <p>Hello Students! Welcome to SIES College.</p>
   <button id="b1">Slide Up</button>
   <button id="b2">Slide Down</button>
</body>
</html>
```

Output:





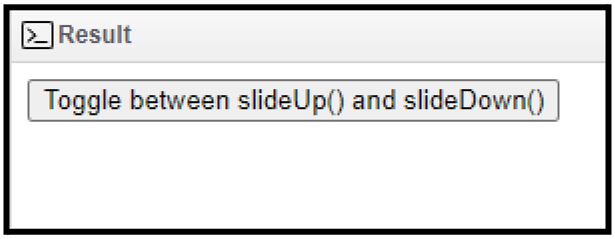
```
Hello Students! Welcome to SIES College.
```

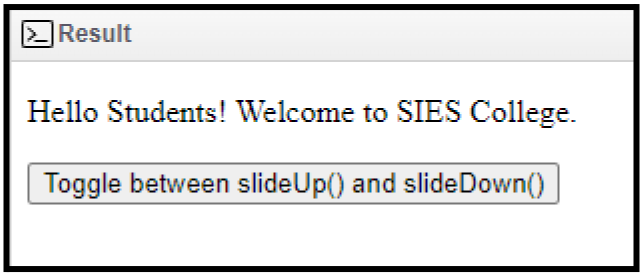
With toggle

Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        $("#b1").click(function(){
            $("p").slideToggle();
        });
    });
</script>
</head>
<body>
   <p>Hello Students! Welcome to SIES College.</p>
   <button id="b1">Toggle between slideUp() and slideDown()</button>

</body>
</html>
```

Output:





Hello Students! Welcome to SIES College.

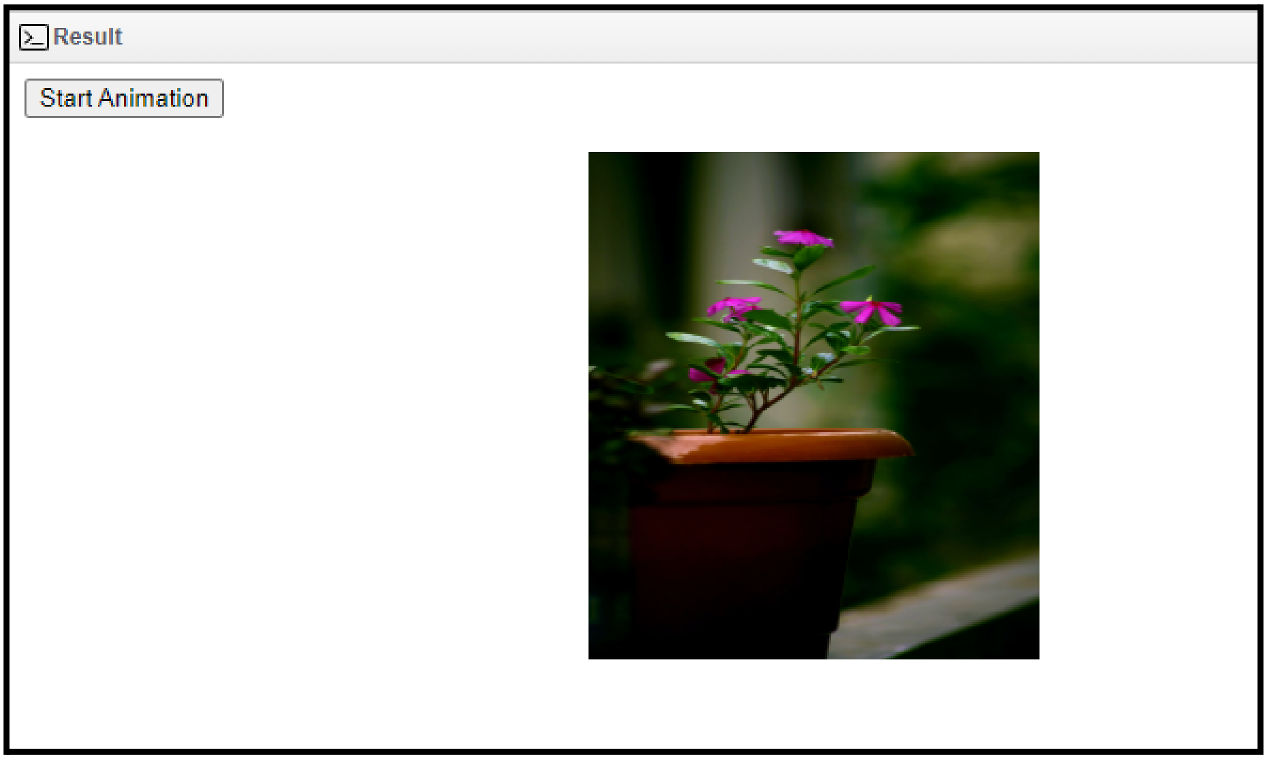
9 jQuery Advanced

9.a.1) jQuery Animation effects, jQuery Chaining

Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<style>
    img{
        position:relative;
    }
</style>

<script>
    $(document).ready(function() {
      $("button").click(function(){
         $("img").animate({left: 300});
      });
  });
</script>
</head>
<body>
  <button>Start Animation</button><br/><br/>
  <img src="https://images.pexels.com/photos/1114112/pexels-photo-1114112.jpeg?cs=srgb&dl=pexels-the-name-is-karupu-1114112.jpg&fm=jpg" alt="jQuery" height="270px" width="240px" />
</body>
</html>
```

Output:

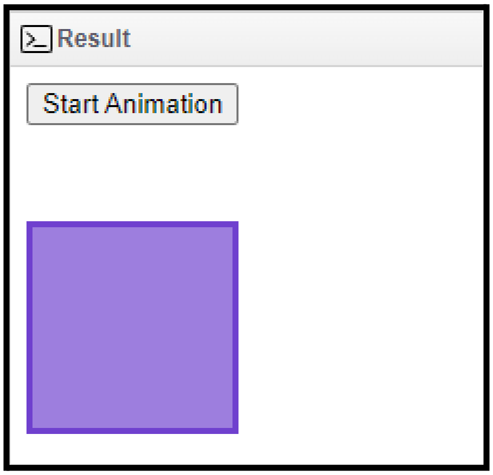


9.a.2)
Code:

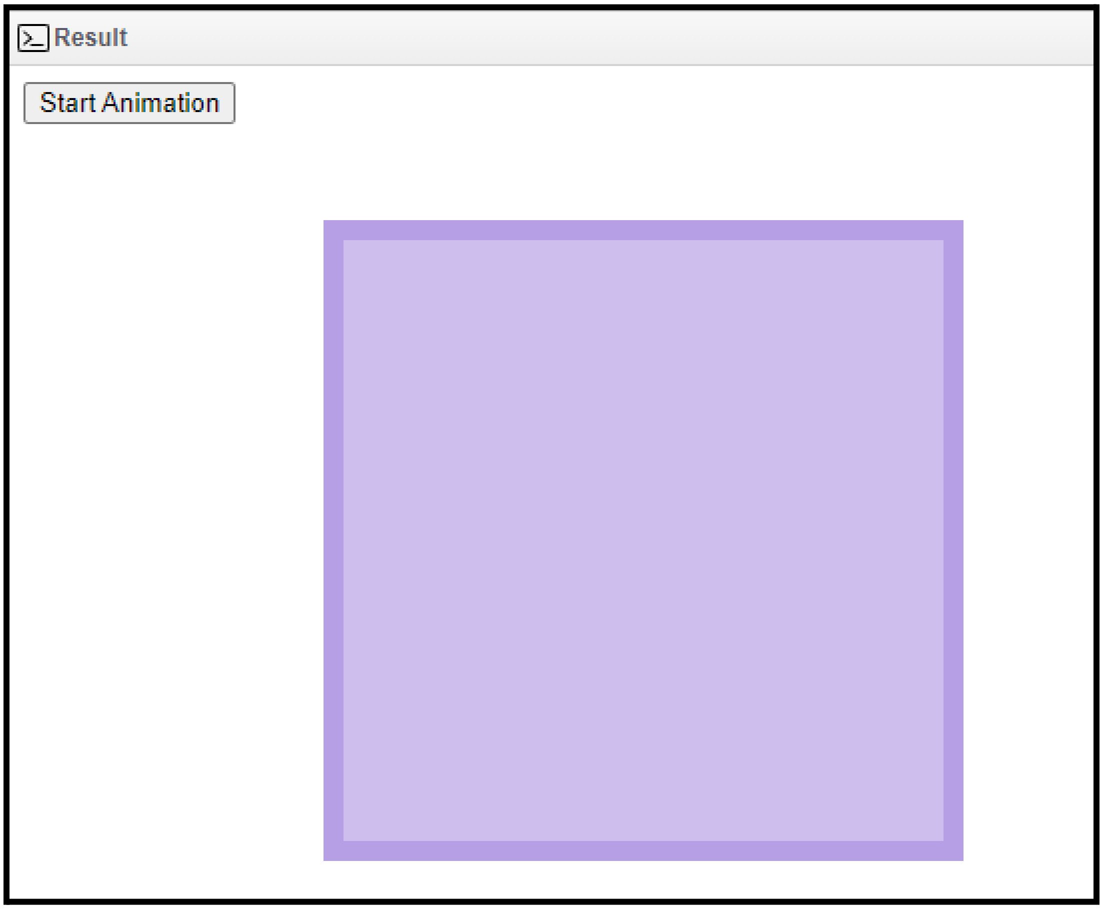
```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
```

```
<style type="text/css">
    .box{
        width:100px;
        height:100px;
        background: #9d7ede;
        margin-top: 30px;
        border-style: solid;
        border-color: #6f40ce;
    }
</style>

<script>
    $(document).ready(function() {
        $("button").click(function(){
            $(".box").animate({
                width: "300px",
                height: "300px",
                marginLeft: "150px",
                borderWidth: "10px",
                opacity:0.5,
            });
        });
   });
</script>
</head>
<body>
   <button type="button">Start Animation</button><br/><br/>
   <div class="box"></div>
</body>
</html>
```

Output: BEFORE



AFTER



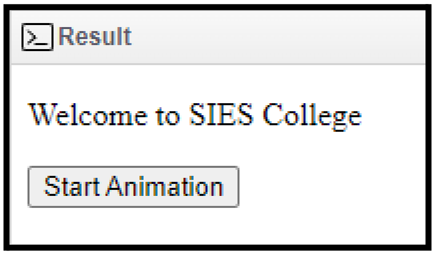
9.a.3) Write Jquery to perform Method Chaining.

Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
   $(document).ready(function() {
      $("button").click(function(){
         $("p").css("color", "pink").slideUp(2000).slideDown(2000);
      });
  });
</script>
</head>
<body>
   <p>Welcome to SIES College</p>
  <button>Start Animation</button><br/><br/>
</body>
```

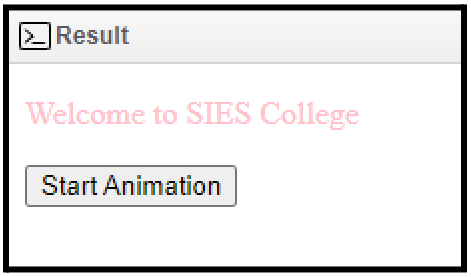
```
</html>
```

Output:
BEFORE



Welcome to SIES College

AFTER



Welcome to SIES College

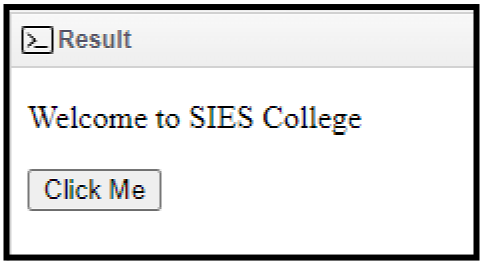
b) jQuery Callback, jQuery Get and Set Contents

1) Write a jquery effect method with a callback function.

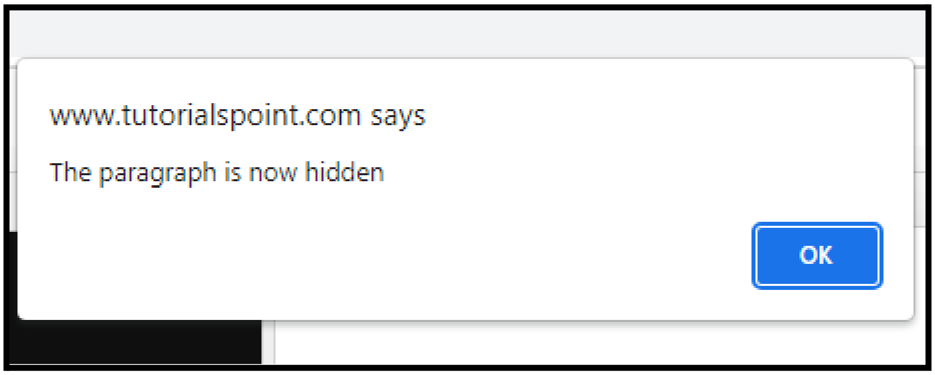
Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        $("button").click(function(){
            $("p").hide("slow", function(){
                alert("The paragraph is now hidden");
            });
        });
   });
</script>
</head>
<body>
    <p>Welcome to SIES College</p>
   <button>Click Me</button><br/><br/>
</body>
</html>
```

Output:



Welcome to SIES College



www.tutorialspoint.com says

The paragraph is now hidden

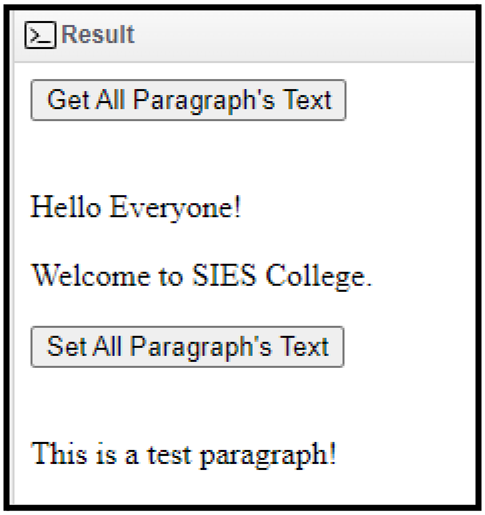
ii) write jquery to get and set text contents of the elements

Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
   $(document).ready(function() {
      $(".b1").click(function(){
         var str = $("p").text();
         alert(str);
         });

         $(".b2").click(function(){
            $("p").text("This is demo text");
      });
  });
</script>
</head>
<body>
  <button class="b1">Get All Paragraph's Text</button><br/><br/>
  <p>Hello Everyone!</p>
  <p>Welcome to SIES College. </p>
```

```
   <button class="b2">Set All Paragraph's Text</button><br/><br/>
   <p>This is a test paragraph!</p>
</body>
</html>
```

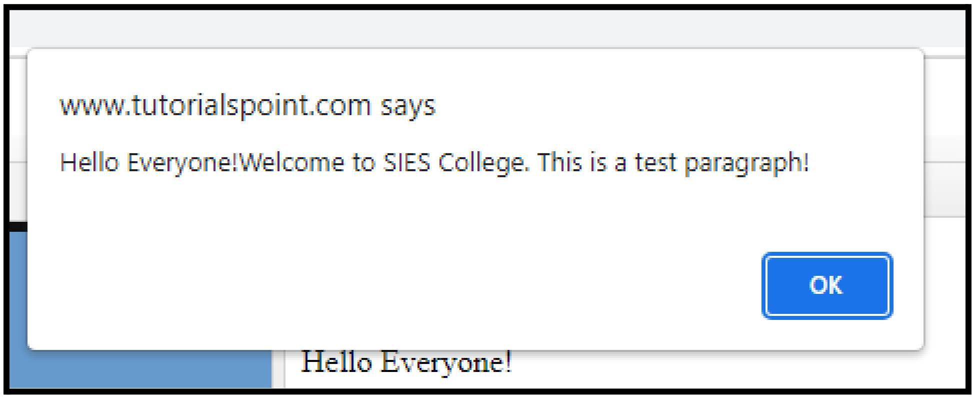
Output:



Hello Everyone!

Welcome to SIES College.

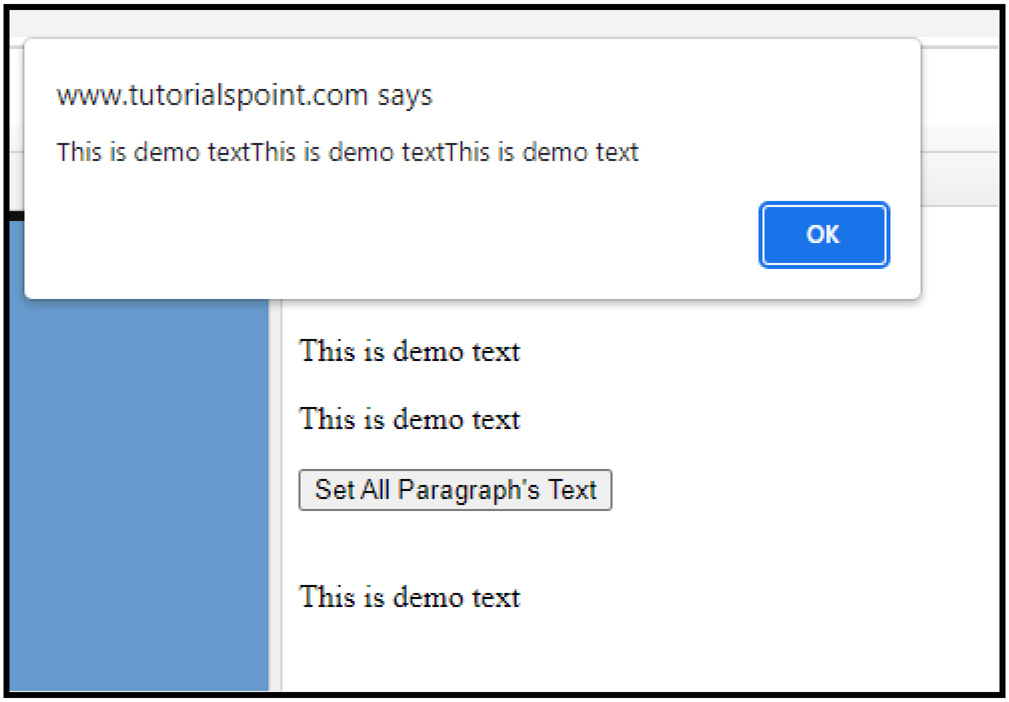
This is a test paragraph!



www.tutorialspoint.com says

Hello Everyone!Welcome to SIES College. This is a test paragraph!

Hello Everyone!



www.tutorialspoint.com says

This is demo textThis is demo textThis is demo text

This is demo text

This is demo text

This is demo text

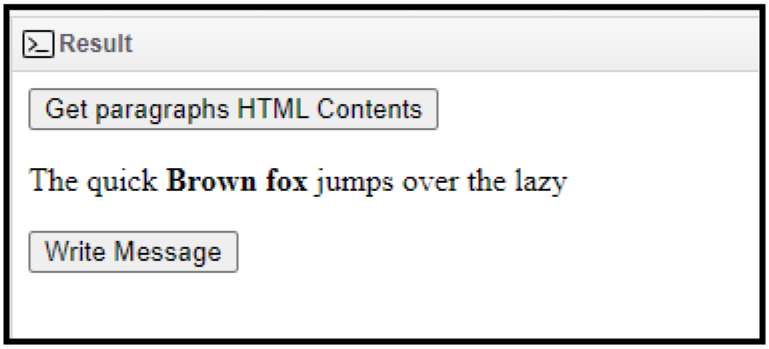
<!doctype html>

III) write a jquery to get and set the html contents of the element..

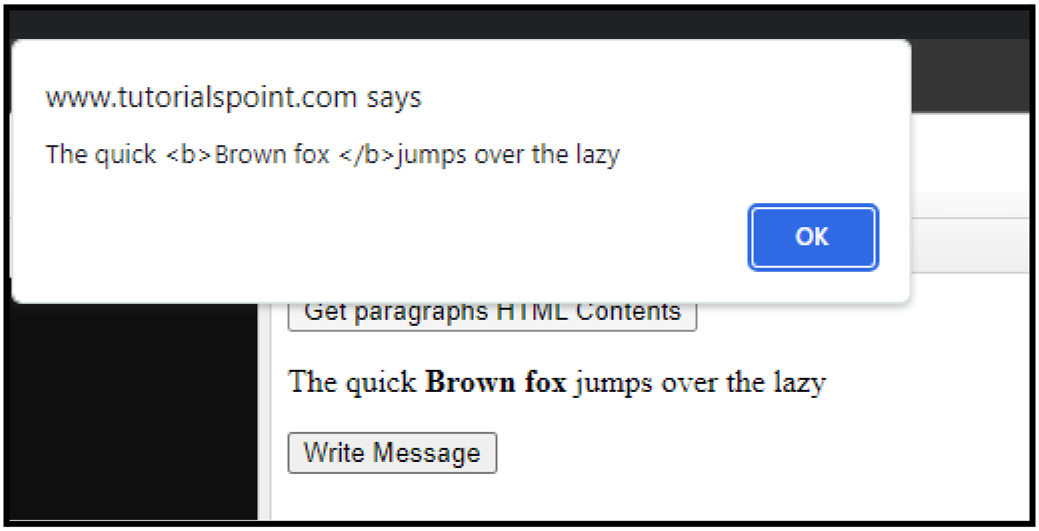
Code:

```
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
    $(".b1").click(function(){
       var str=$("p").html();
       alert(str);
    });
    $(".b2").click(function(){
        $("body").html("<p>hello world</p>");
     });
  });
</script>
</head>
<body>
   <button class="b1">Get paragraphs HTML Contents</button>
   <p>The quick <b>Brown fox </b>jumps over the lazy</p>
   <button class="b2">Write Message</button>
</body>
</html>
```

Output:



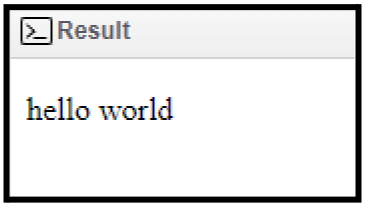
The quick **Brown fox** jumps over the lazy



www.tutorialspoint.com says

The quick <b>Brown fox </b>jumps over the lazy

The quick **Brown fox** jumps over the lazy



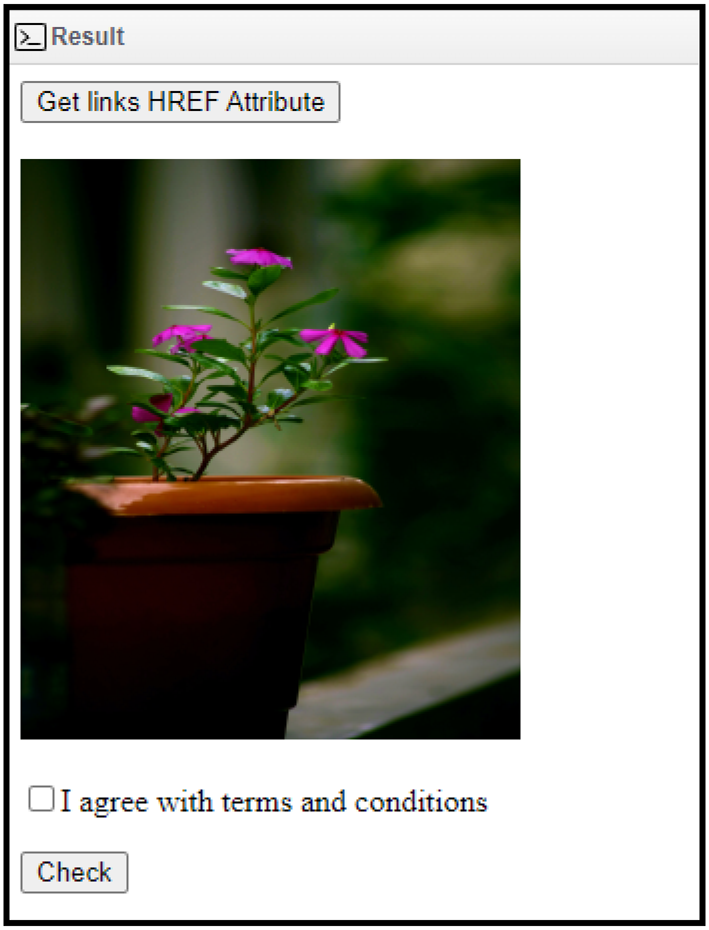
hello world

4) Write a jquery to get and set the value of the attribute of an element
Code:

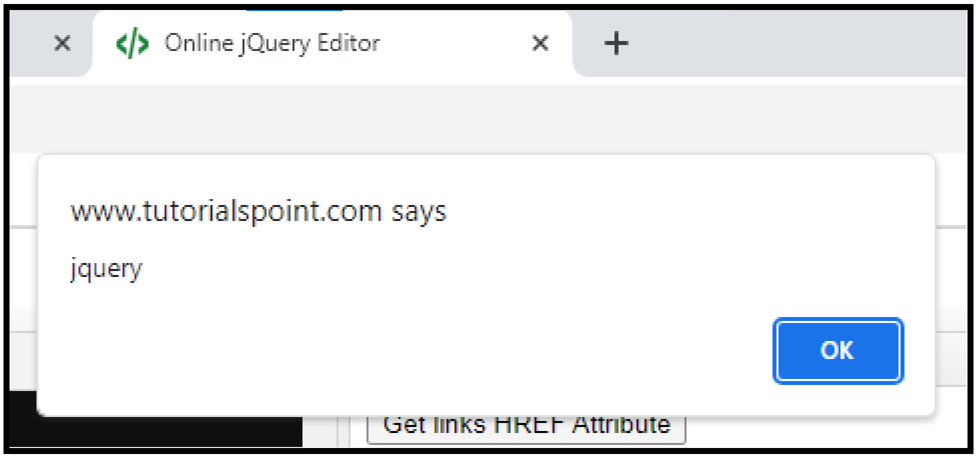
```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
     $(".b1").click(function(){
        var str = $("img#jquery").attr("alt");
```

```
      alert(str);
    });
    $(".b2").click(function(){
      $('input[type="checkbox"]').attr("checked","checked");
    });
  });
</script>
</head>
<body>
  <button class="b1">Get links HREF Attribute</button><br/><br/>
  <img id="jquery"
src="https://images.pexels.com/photos/1114112/pexels-photo-1114112.jpeg?cs=srgb&dl=pexels-the-name-is-karupu-1114112.jpg&fm=jpg" alt="jquery" height="290px" width="250px"/><br />
  <p><input type="checkbox"/>I agree with terms and conditions</p>
  <button class="b2">Check</button>
</body>
</html>
```

Output:

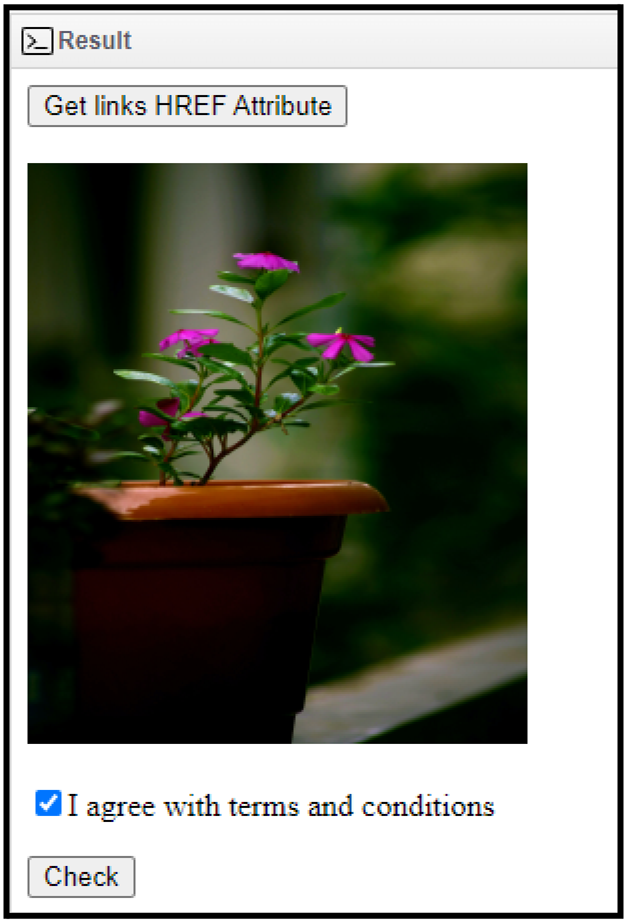


□ I agree with terms and conditions



www.tutorialspoint.com says

jquery



☑ I agree with terms and conditions

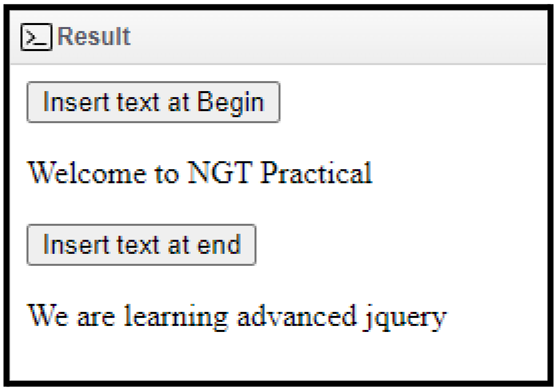
C. jQuery Insert Content, jQuery Remove Elements and Attribute. 1) Write a jquery to insert HTML elements at the beginning and end of the elements.

Code:

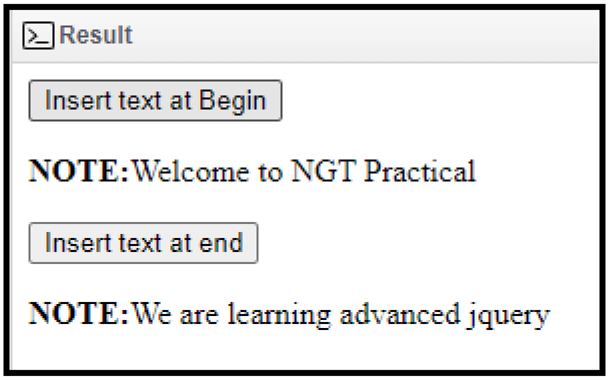
```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
```

```
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
      $(".b1").click(function(){
          $("p").prepend("<strong>NOTE:</strong>");
      });
      $(".b2").click(function(){
          $("#container").append("this is demo text.");
      });
  });
</script>
</head>
<body>
    <button class="b1">Insert text at Begin</button>
    <p>Welcome to NGT Practical</p>
    <button class="b2">Insert text at end</button>
    <div id="container">
        <p>We are learning advanced jquery</p>
    </div>
</body>
</html>
```

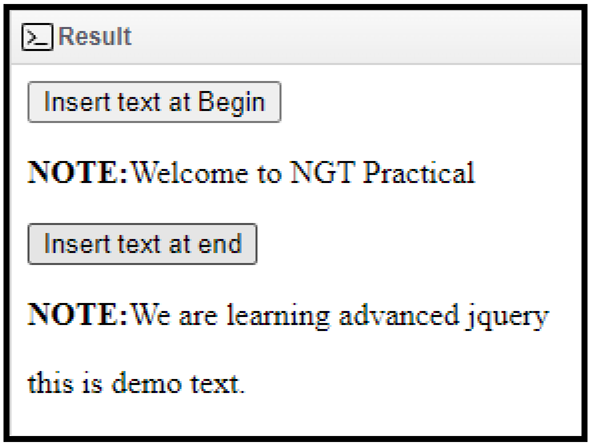
Output:



```
Welcome to NGT Practical

We are learning advanced jquery
```



```
NOTE:Welcome to NGT Practical

NOTE:We are learning advanced jquery
```



NOTE:Welcome to NGT Practical

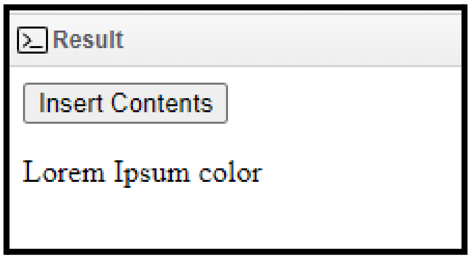
NOTE:We are learning advanced jquery

this is demo text.

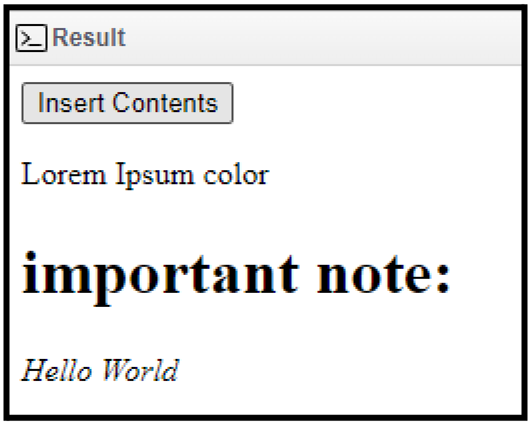
2) Write a jquery to insert multiple HTML elements at the beginning and end of the elements.
Code:

```
<!doctype html>
<html>
<head></head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
      $(".b1").click(function(){
          var newheading= "<h1> important note:</h1>";
          var newparagraph=document.createElement("p");
          newparagraph.innerHTML="<em>Hello World</em>";
          $("body").append(newheading,newparagraph);
      });
  });
</script>
</head>
<body>
   <button class="b1">Insert Contents</button>
   <p>Lorem Ipsum color </p>
</body>
</html>
```

Output:



Lorem Ipsum color



Lorem Ipsum color

important note:

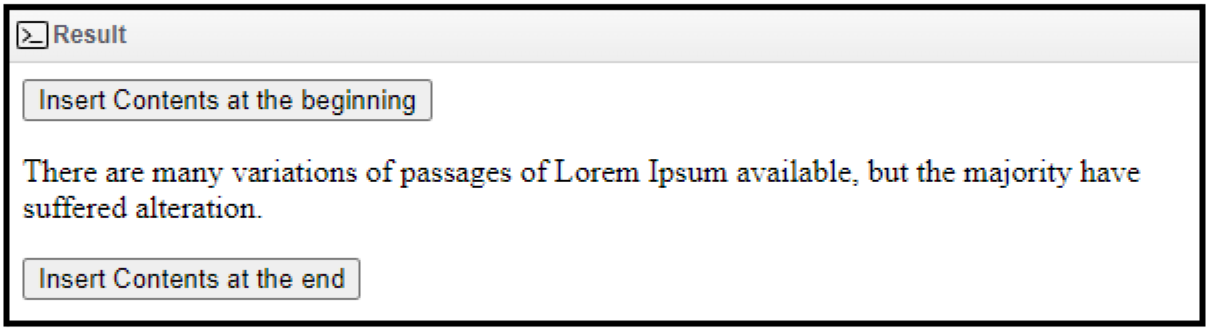
Hello World

3) Write a jquery to insert multiple HTML elements before and after the elements

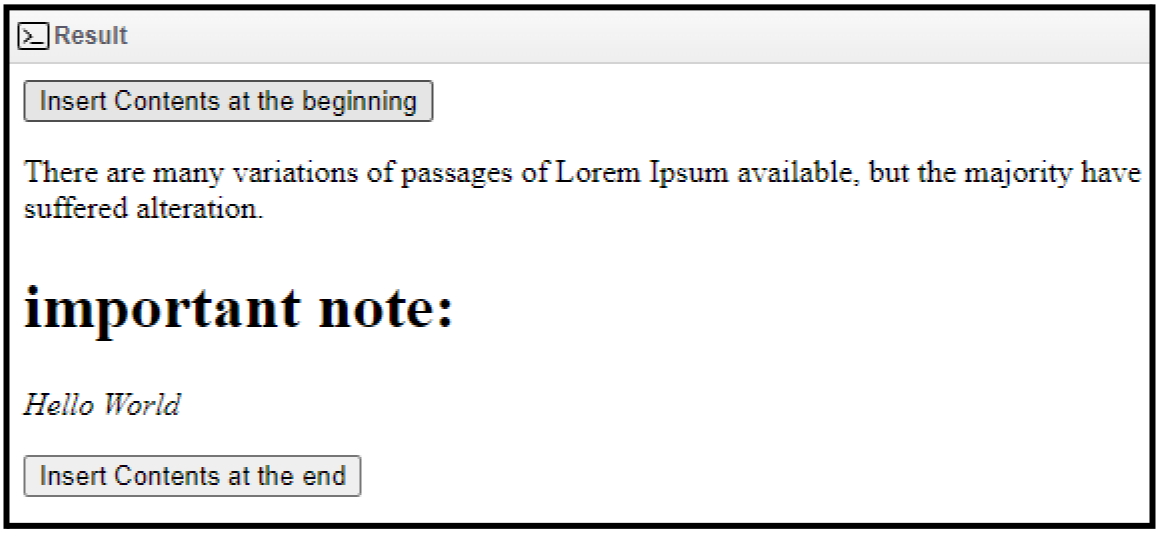
Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
      $(".b1").click(function(){
          var newheading= "<h1> important note:</h1>";
          var newparagraph=document.createElement("p");
          newparagraph.innerHTML="<em>Hello World</em>";
          $("p").append(newheading,newparagraph);
      });
      $(".b2").click(function(){
          var newheading= "<h1> important note:</h1>";
          var newparagraph=document.createElement("p");
          newparagraph.innerHTML="<em>Welcome</em>";
          $("body").append(newheading,newparagraph);
      });
  });
</script>
</head>
<body>
   <button class="b1">Insert Contents at the beginning</button>
   <p>There are many variations of passages of Lorem Ipsum available, but the majority have
suffered alteration. </p>
   <button class="b2">Insert Contents at the end</button>
</body>
</html>
```

Output:



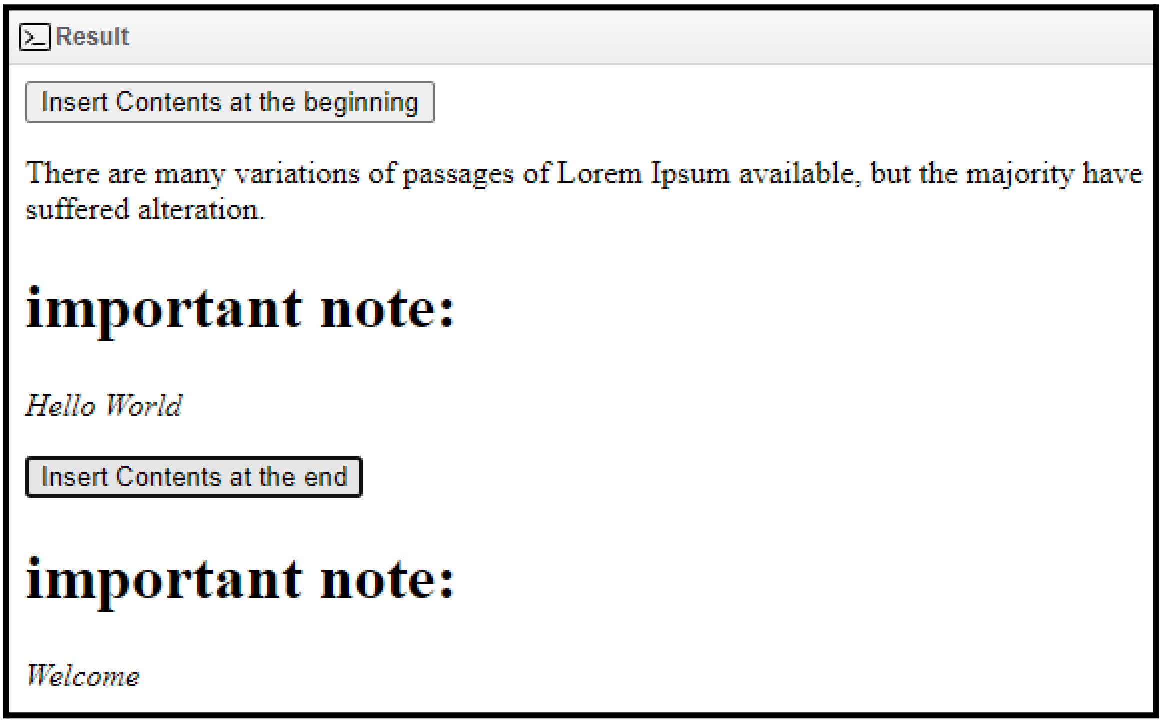
There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration.



There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration.

important note:

Hello World



There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration.

important note:

Hello World

important note:

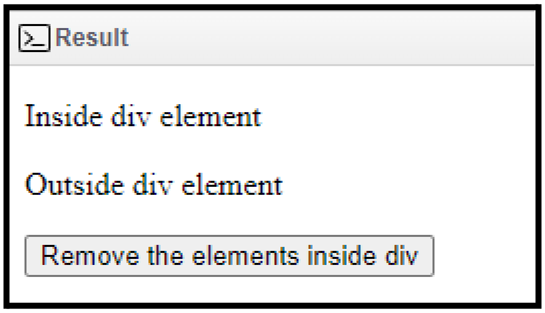
Welcome

4) Write a jquery to remove the contents of the elements.

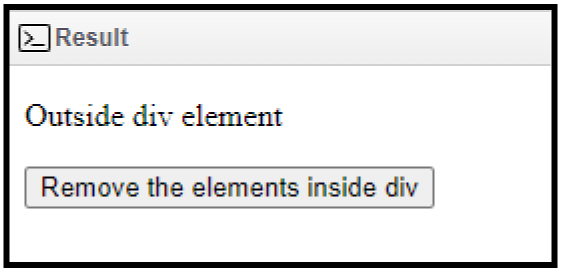
Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
      $(".btn").click(function(){
          $("div#demo").remove();
      });
  });
</script>
</head>
<body>
    <div id="demo">
        <p>Inside div element</p>
    </div>
    <p>Outside div element</p>
    <button class="btn">Remove the elements inside div</button>
</body>
</html>
```

Output:



```
Inside div element

Outside div element
```



```
Outside div element
```

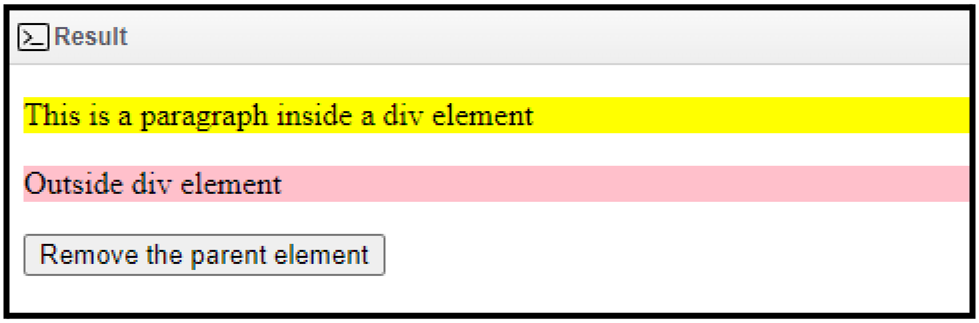
5) Write a jquery to remove the parent element of an HTML element from the page

Code:

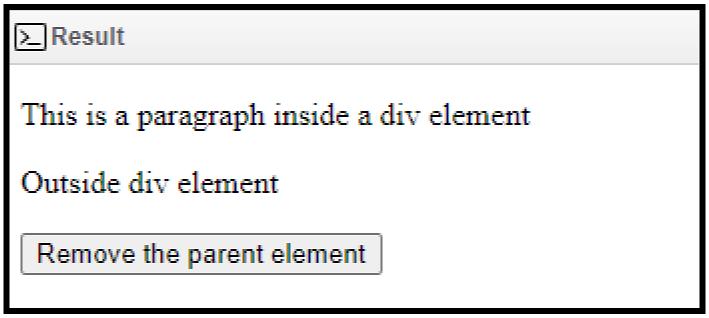
```
<!doctype html>
```

```
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
   $(document).ready(function() {
       $(".btn").click(function(){
           $("p").unwrap();
       });
   });
</script>
<style>
div{
    background-color:yellow;
}
article{
    background-color:pink;
}
</style>
</head>
<body>
    <div>
        <p>This is a paragraph inside a div element</p>
    </div>
    <article>
        <p>Outside div element</p>
    </article>
    <button class="btn">Remove the parent element</button>
</body>
</html>
```

Output:



```
This is a paragraph inside a div element

Outside div element
```



This is a paragraph inside a div element

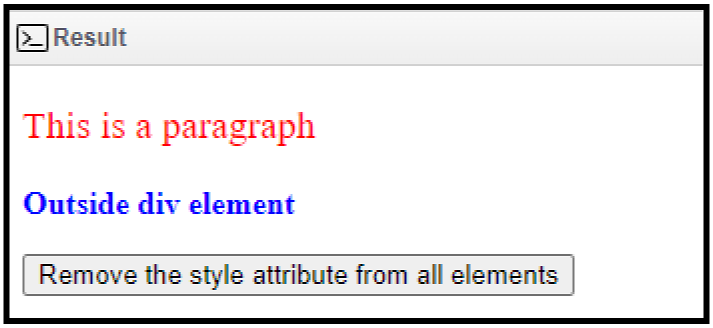
Outside div element

6) Write a jquery to remove an attribute from the HTML elements

Code:

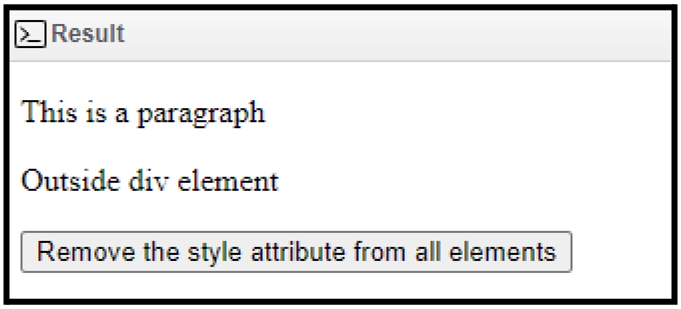
```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
     $(".btn").click(function(){
        $("p").removeAttr("style");
     });
  });
</script>
<style>
div{
   background-color:yellow;
}
article{
   background-color:pink;
}
</style>
</head>
<body>
   <p style="font-size:120%; color:red">This is a paragraph</p>
   <p style="font-weight:bold;color:blue">Outside div element</p>
   <button class="btn">Remove the style attribute from all  elements</button>
</body>
</html>
```

Output:



This is a paragraph

Outside div element



This is a paragraph

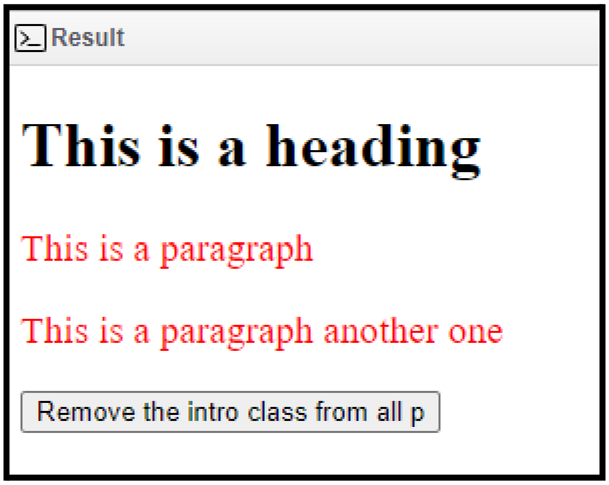
Outside div element

7) Write a jquery to add and remove CSS classes from the HTML elements
Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
     $(".btn").click(function(){
        $("p").removeClass("intro");
     });
  });
</script>
<style>
.intro{
   font-size:120%;
   color:red;
}
</style>
</head>
<body>
   <h1>This is a heading</h1>
   <p class="intro">This is a paragraph</p>
   <p class="intro">This is a paragraph another one</p>
   <button class="btn">Remove the intro class from all p</button>
```

```
</body>
</html>
```

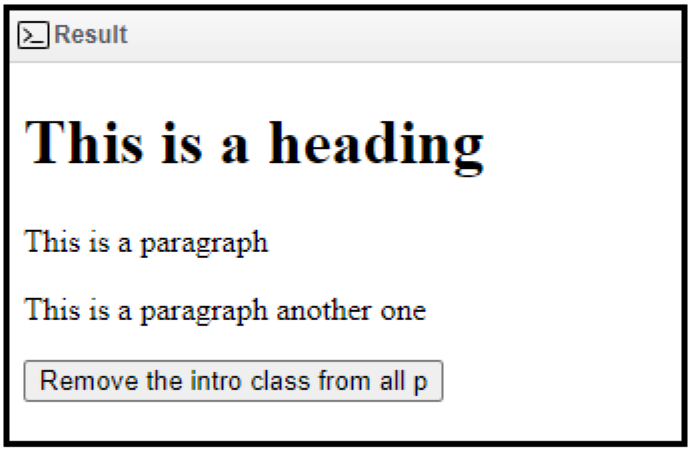
Output:



This is a heading

This is a paragraph

This is a paragraph another one



This is a heading

This is a paragraph

This is a paragraph another one

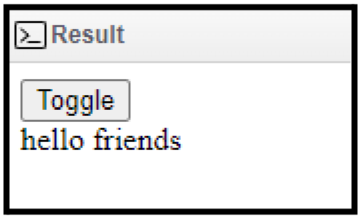
8) Write a jquery to set the duration in slide toggle effect.

Code:

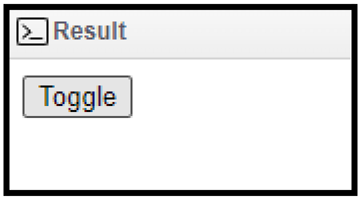
```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function() {
      $(".b1").click(function(){
          $(".box").slideToggle();
      })
  });
</script>
</head>
```

```
<body>
    <button type="button" class="b1">Toggle</button>
    <div class="box">
        <div class="box-inner">
            hello friends
        </div>
    </div>
</body>
</html>
```

Output:



hello friends



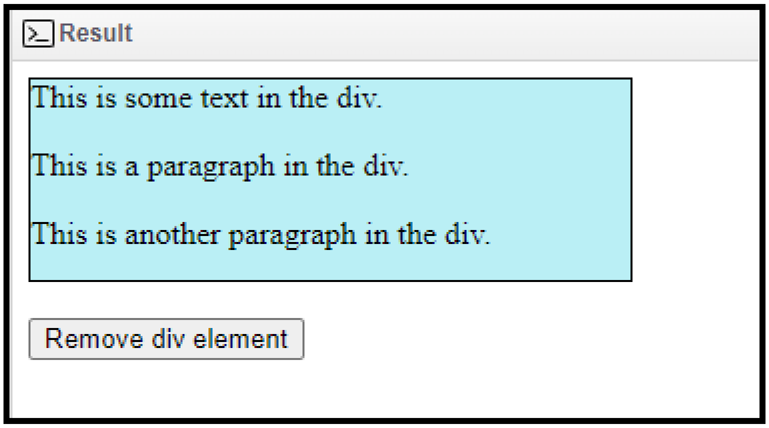
9) Write a jquery to remove the HTML elements from the page.

Code:

```
<!doctype html>
<html>
<head>
<title>Online jQuery Editor</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
   $(document).ready(function() {
      $("button").click(function(){
          $("#div1").remove();
      });
   });
</script>
</head>
<body>
   <div id="div1"
   style="
   height:100px;
   width:300px;
   border: 1px solid black;
```

```
  background-color: #bcf0f5;
  ">
     This is some text in the div.
     <p>This is a paragraph in the div.</p>
     <p>This is another paragraph in the div.</p>
  </div>
  <br/>
  <button>Remove div element</button>
</body>
</html>
```

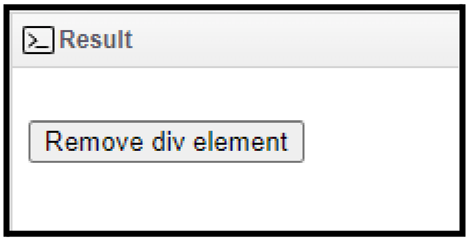
Output:



This is some text in the div.

This is a paragraph in the div.

This is another paragraph in the div.



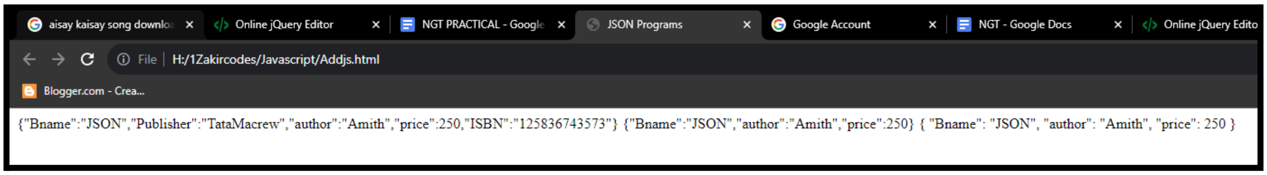
10 JSON

a Creating JSON

Code:

```
<html>
<head>
  <title>JSON Programs</title>
</head>
<body>
   <script type="text/javascript">
      var data={
         Bname: "JSON",
         Publisher:"TataMacrew",
         author:"Amith",
            price:250,
            ISBN: "125836743573",
        };
        document.writeln(JSON.stringify(data));
        document.writeln(JSON.stringify(data, ["Bname","author","price"]));
        document.writeln(JSON.stringify(data,["Bname", "author", "price"], 5));
    </script>
</body>
</html>
```

Output:



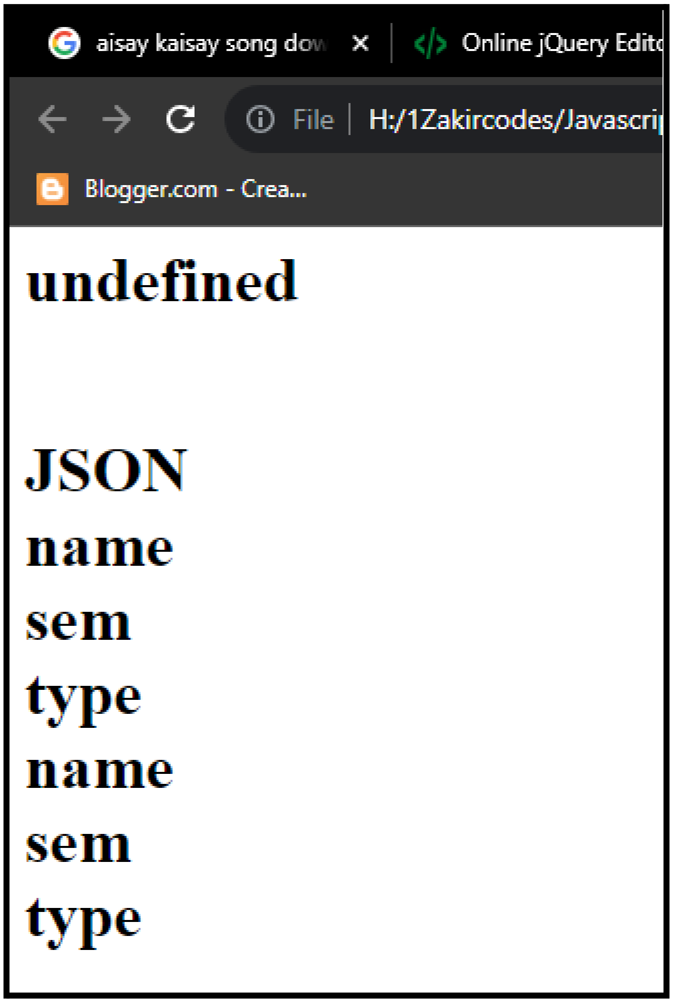
```
{"Bname":"JSON","Publisher":"TataMacrew","author":"Amith","price":250,"ISBN":"125836743573"} {"Bname":"JSON","author":"Amith","price":250} { "Bname": "JSON", "author": "Amith", "price": 250 }
```

b Parsing JSON

Code:

```
<html>
<head>
   <title>JSON Programs</title>
</head>
<body>
    <script type="text/javascript">
        var data='{"name": "JSON","sem": 5,"type":"programming"}';
            document.writeln("<h1>"+ data.name + "<h1><br/>");
                var json_obj = JSON.parse(data, function (name,value){
                    return value;
                });
        document.writeln(json_obj.name);
        for(key in json_obj){
            document.write("<br/>" + key);
        }
        var json_obj = JSON.parse(data, function (name,value){
            return value;
        document.writeln(json_obj.name);
        if(name=="sem"){
        return undefined;
        }
        else{
            return value;
        }
        });
        for(key in json_obj){
            document.write("<br/>" + key);
        }
    </script>
</body>
</html>
```

Output:



```
undefined


JSON
name
sem
type
name
sem
type
```

Pymongo
Install using pip
Pip install pymongo