# INDEX

## Subject – Next Generation Databases Practical

| Sr. No. | Aim | Date | Teacher's Sign |
|---|---|---|---|
| 1 | **MongoDB Basics**<br>a. Write a MongoDB query to create and drop database.<br>b. Write a MongoDB query to create, display and drop collection<br>c. Write a MongoDB query to insert, query, update and delete a document. | | |
| 2 | **Implementing Aggregation**<br>a. Write a MongoDB query to use sum, avg, min and max expression.<br>b. Write a MongoDB query to use push and addToSet expression.<br>c. Write a MongoDB query to use first and last expression. | | |
| 3 | **Programs on Basic jQuery**<br>a. jQuery Basic, jQuery Events<br>b. jQuery Selectors, jQuery Hide and Show effects<br>c. jQuery fading effects, jQuery Sliding effects | | |
| 4 | **jQuery Advanced**<br>a. jQuery Chaining | | |
| 5 | **JSON**<br>a. Creating JSON<br>b. Parsing JSON<br>c. Persisting JSON | | |
| 6 | **MongoDB and JSON**<br>a. Create a JSON file and import it to MongoDB<br>b. Export MongoDB to JSON. | | |

# Practical No.: 1

## MongoDB Basics

### a. Write a MongoDB query to create and drop database.

**Syntax:**

Create – use database_name

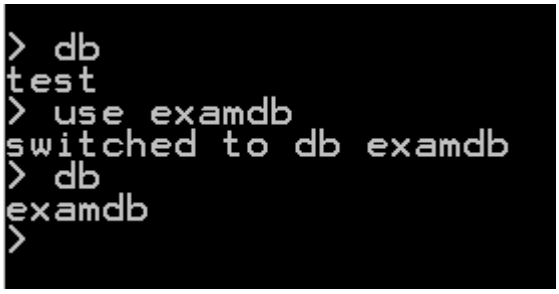
Drop – db.dropDatabase()

**Source Code:**

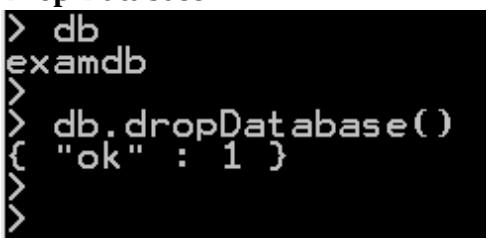
```
use examdb
db.dropDatabase()
```

**Output:**

**Create Database –**




**Drop Database –**



## b. Aim: Write a MongoDB query to create, display and drop collection.

**Syntax:**

**Create –**

db.createCollection(collection_name)

**Display –**

Show collections

**Drop –**

db.collection_name.drop()

**Source Code:**

```
use examdb
db.createCollection("student")
show collections
db.student.drop()
```

**Output:**

**Create –**

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

**Display –**

```
>
> show collections
student
>
>
```

**Drop –**

```
>
> db.student.drop()
true
>
> show collections
>
>
```

## c. Write a MongoDB query to insert, query, update and delete a document.

**Syntax:**

**To Insert Document –**

db.COLLECTION_NAME.insert(document)

**To Query Document –**

db.COLLECTION_NAME.find()

**To Update Document –**

db.COLLECTION_NAME.update(SELECTION_CRITERIA,UPDATED DATA)

**To Delete Document –**

db.COLLECTION_NAME.remove(DELETION_CRITERIA)

**Source Code –**

```
use examdb

db.createCollection("student")

db.student.insert({Name:"S1", Gender: "M", Class: "C1", Score: 95, Age: 25})
db.student.insert({Name:"S2", Gender: "M", Class: "C1", Score: 85, Age: 18})

var d = [
{Name:"S3", Gender: "F", Class: "C1", Score: 85, Age: 18},
{Name:"S4", Gender: "F", Class: "C1", Score: 75, Age: 18},
{Name:"S5", Gender: "F", Class: "C2", Score: 75, Age: 18},
{Name:"S6", Gender: "M", Class: "C2", Score: 100, Age: 21},
{Name:"S7", Gender: "M", Class: "C2", Score: 100, Age: 21},
{Name:"S8", Gender: "F", Class: "C2", Score: 100, Age: 25},
{Name:"S9", Gender: "F", Class: "C2", Score: 90, Age: 25},
{Name:"S10", Gender: "F", Class: "C3", Score: 90, Age: 28},
{Name:"Student1", Gender: "M", Class: "Biology", Score: 90, Age: 30},
{Name:"Student2", Gender: "M", Class: "Chemistry", Score: 90, Age: 30},
{Name:"Test1", Gender: "M", Class: "Chemistry", Score: 90, Age: 30},
{Name:"Test2", Gender: "M", Class: "Chemistry", Score: 90, Age: 30},
{Name:"Test3", Gender: "M", Class: "Chemistry", Score: 90, Age: 30},
{Name:"Test4", Gender: "F", Class: "Chemistry", Score: 90, Age: 30},
];
db.student.insert(d)

db.student.find()

db.student.update({name:"S1"},{$set:{name:"MSC_DS"}})
db.stdent.find()

db.student.remove({name:"Test4"})

db.student.find()
```

**Output:**

**Insert Document –**

```
> use examdb
switched to db examdb
> db.createCollection("student")
{ "ok" : 1 }
> db.student.insert({Name:"S1", Gender: "M", Class: "C1", Score: 95, Age: 25})
WriteResult({ "nInserted" : 1 })
> db.student.insert({Name:"S2", Gender: "M", Class: "C1", Score: 85, Age: 18})
WriteResult({ "nInserted" : 1 })
>
```

```
> var d = [
... {Name:"S3", Gender: "F", Class: "C1", Score: 85, Age: 18},
... {Name:"S4", Gender: "F", Class: "C1", Score: 75, Age: 18},
... {Name:"S5", Gender: "F", Class: "C2", Score: 75, Age: 18},
... {Name:"S6", Gender: "M", Class: "C2", Score: 100, Age: 21},
... {Name:"S7", Gender: "M", Class: "C2", Score: 100, Age: 21},
... {Name:"S8", Gender: "F", Class: "C2", Score: 100, Age: 25},
... {Name:"S9", Gender: "F", Class: "C2", Score: 90, Age: 25},
... {Name:"S10", Gender: "F", Class: "C3", Score: 90, Age: 28},
... {Name:"Student1", Gender: "M", Class: "Biology", Score: 90, Age: 30},
... {Name:"Student2", Gender: "M", Class: "CHemistry", Score: 90, Age: 30},
... {Name:"Test1", Gender: "M", Class: "Chemistry", Score: 90, Age: 30},
... {Name:"Test2", Gender: "M", Class: "Chemistry", Score: 90, Age: 30},
... {Name:"Test3", Gender: "M", Class: "Chemistry", Score: 90, Age: 30},
... {Name:"Test4", Gender: "F", Class: "Chemistry", Score: 90, Age: 30},
... ];
> db.student.insert(d)
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 14,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
>
>
```

**Query Document –**

```
> db.student.find()
{ "_id" : ObjectId("6470d24f7a78053aa65a7a56"), "Name" : "S1", "Gender" : "M", "Class" : "C1", "Score" : 95, "Age" : 25 }
{ "_id" : ObjectId("6470d26f7a78053aa65a7a57"), "Name" : "S2", "Gender" : "M", "Class" : "C1", "Score" : 85, "Age" : 18 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a58"), "Name" : "S3", "Gender" : "F", "Class" : "C1", "Score" : 85, "Age" : 18 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a59"), "Name" : "S4", "Gender" : "F", "Class" : "C1", "Score" : 75, "Age" : 18 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5a"), "Name" : "S5", "Gender" : "F", "Class" : "C2", "Score" : 75, "Age" : 18 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5b"), "Name" : "S6", "Gender" : "M", "Class" : "C2", "Score" : 100, "Age" : 21 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5c"), "Name" : "S7", "Gender" : "M", "Class" : "C2", "Score" : 100, "Age" : 21 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5d"), "Name" : "S8", "Gender" : "F", "Class" : "C2", "Score" : 100, "Age" : 25 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5e"), "Name" : "S9", "Gender" : "F", "Class" : "C2", "Score" : 90, "Age" : 25 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5f"), "Name" : "S10", "Gender" : "F", "Class" : "C3", "Score" : 90, "Age" : 28 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a60"), "Name" : "Student1", "Gender" : "M", "Class" : "Biology", "Score" : 90, "Age" : 30 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a61"), "Name" : "Student2", "Gender" : "M", "Class" : "Chemistry", "Score" : 90, "Age" : 30 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a62"), "Name" : "Test1", "Gender" : "M", "Class" : "Chemistry", "Score" : 90, "Age" : 30 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a63"), "Name" : "Test2", "Gender" : "M", "Class" : "Chemistry", "Score" : 90, "Age" : 30 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a64"), "Name" : "Test3", "Gender" : "M", "Class" : "Chemistry", "Score" : 90, "Age" : 30 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a65"), "Name" : "Test4", "Gender" : "F", "Class" : "Chemistry", "Score" : 90, "Age" : 30 }
>
>
```

**Update Document –**

```
>
> db.student.update({Name:"S1"},{$set:{Name:"MSC_DS"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

```
>
> db.student.find()
{ "_id" : ObjectId("6470d24f7a78053aa65a7a56"), "Name" : "MSC_DS", "Gender" : "M", "Class" : "C1", "Score" : 95, "Age" : 25 }
{ "_id" : ObjectId("6470d26f7a78053aa65a7a57"), "Name" : "S2", "Gender" : "M", "Class" : "C1", "Score" : 85, "Age" : 18 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a58"), "Name" : "S3", "Gender" : "F", "Class" : "C1", "Score" : 85, "Age" : 18 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a59"), "Name" : "S4", "Gender" : "F", "Class" : "C1", "Score" : 75, "Age" : 18 }
```

**Delete Document –**

```
>
> db.student.remove({name:"Test4"})
WriteResult({ "nRemoved" : 0 })
>
```

```
>
> db.student.find()
{ "_id" : ObjectId("6470d24f7a78053aa65a7a56"), "Name" : "MSC_DS", "Gender" : "M", "Class" : "C1", "Score" : 95, "Age" : 25 }
{ "_id" : ObjectId("6470d26f7a78053aa65a7a57"), "Name" : "S2", "Gender" : "M", "Class" : "C1", "Score" : 85, "Age" : 18 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a58"), "Name" : "S3", "Gender" : "F", "Class" : "C1", "Score" : 85, "Age" : 18 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a59"), "Name" : "S4", "Gender" : "F", "Class" : "C1", "Score" : 75, "Age" : 18 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5a"), "Name" : "S5", "Gender" : "F", "Class" : "C2", "Score" : 75, "Age" : 18 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5b"), "Name" : "S6", "Gender" : "M", "Class" : "C2", "Score" : 100, "Age" : 21 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5c"), "Name" : "S7", "Gender" : "M", "Class" : "C2", "Score" : 100, "Age" : 21 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5d"), "Name" : "S8", "Gender" : "F", "Class" : "C2", "Score" : 100, "Age" : 25 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5e"), "Name" : "S9", "Gender" : "F", "Class" : "C2", "Score" : 90, "Age" : 25 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a5f"), "Name" : "S10", "Gender" : "F", "Class" : "C3", "Score" : 90, "Age" : 28 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a60"), "Name" : "Student1", "Gender" : "M", "Class" : "Biology", "Score" : 90, "Age" : 30 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a61"), "Name" : "Student2", "Gender" : "M", "Class" : "Chemistry", "Score" : 90, "Age" : 30 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a62"), "Name" : "Test1", "Gender" : "M", "Class" : "Chemistry", "Score" : 90, "Age" : 30 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a63"), "Name" : "Test2", "Gender" : "M", "Class" : "Chemistry", "Score" : 90, "Age" : 30 }
{ "_id" : ObjectId("6470d6c47a78053aa65a7a64"), "Name" : "Test3", "Gender" : "M", "Class" : "Chemistry", "Score" : 90, "Age" : 30 }
>
```

## Practical No.: 2

## Aim: Implementing Aggregation

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

**Syntax:**

**Sum**

{$sum:[<expression1>,<expression>…]}

**Avg**

{$avg:[<expression1>…]}

**Min**

{$min:[<expression1>…]}

**Max**

{$Max:[<expression1>…]}

**Source Code:**

**Sum**

```
use examdb
db.student.aggregate([{$group:{_id:"$Class",sum:{$sum:"$Score"}}}])
```

**Avg**

```
use examdb
db.student.aggregate([{$group:{_id:"$Class",Average:{$avg:"$Score"}}}])
```

**Min**

```
use examdb
db.student.aggregate([{$group:{_id:"$Class",Minimum:{$min:"$Score"}}}])
```

**Max**

```
use examdb
db.student.aggregate([{$group:{_id:"$Class",Maximum:{$max:"$Score"}}}])
```

**Output:**

**Sum**

```
> db.student.aggregate([{$group:{_id:"$Class",sum:{$sum:"$Score"}}}])
{ "_id" : "C1", "sum" : 340 }
{ "_id" : "C2", "sum" : 465 }
{ "_id" : "Biology", "sum" : 90 }
{ "_id" : "Chemistry", "sum" : 360 }
{ "_id" : "C3", "sum" : 90 }
>
```

**Avg**

```
> db.student.aggregate([{$group:{_id:"$Class",Average:{$avg:"$Score"}}}])
{ "_id" : "Biology", "Average" : 90 }
{ "_id" : "Chemistry", "Average" : 90 }
{ "_id" : "C3", "Average" : 90 }
{ "_id" : "C1", "Average" : 85 }
{ "_id" : "C2", "Average" : 93 }
>
```

**Min**

```
> db.student.aggregate([{$group:{_id:"$Class",Minimum:{$min:"$Score"}}}])
{ "_id" : "C3", "Minimum" : 90 }
{ "_id" : "C1", "Minimum" : 75 }
{ "_id" : "C2", "Minimum" : 75 }
{ "_id" : "Biology", "Minimum" : 90 }
{ "_id" : "Chemistry", "Minimum" : 90 }
>
```

**Max**

```
> db.student.aggregate([{$group:{_id:"$Class",Maximum:{$max:"$Score"}}}])
{ "_id" : "Chemistry", "Maximum" : 90 }
{ "_id" : "C3", "Maximum" : 90 }
{ "_id" : "C1", "Maximum" : 95 }
{ "_id" : "C2", "Maximum" : 100 }
{ "_id" : "Biology", "Maximum" : 90 }
>
```

**b. Write a MongoDB query to use push and addToSet expression.**

## 1. addToSet:

**Syntax:**

{$addToSet:<expression>}

**Source Code:**

```
use examdb
db.student.aggregate([{$group:{_id:"$Class",AddToSet:{$addToSet:"$Score"}}}])
```

**Output:**

```
> db.student.aggregate([{$group:{_id:"$Class",AddToSet:{$addToSet:"$Score"}}}])
{ "_id" : "C3", "AddToSet" : [ 90 ] }
{ "_id" : "C1", "AddToSet" : [ 95, 85, 75 ] }
{ "_id" : "C2", "AddToSet" : [ 90, 100, 75 ] }
{ "_id" : "Biology", "AddToSet" : [ 90 ] }
{ "_id" : "Chemistry", "AddToSet" : [ 90 ] }
>
```

## 2. Push:

**Syntax:**

{$push:<expression>}

**Source Code:**

```
use examdb
db.student.aggregate([{$group:{_id:"$Class",Push:{$push:"$Score"}}}])
```

**Output:**

```
> db.student.aggregate([{$group:{_id:"$Class",Push:{$push:"$Score"}}}])
{ "_id" : "Biology", "Push" : [ 90 ] }
{ "_id" : "Chemistry", "Push" : [ 90, 90, 90, 90 ] }
{ "_id" : "C3", "Push" : [ 90 ] }
{ "_id" : "C1", "Push" : [ 95, 85, 85, 75 ] }
{ "_id" : "C2", "Push" : [ 75, 100, 100, 100, 90 ] }
>
```

**c. Write a MongoDB query to use first and last expression.**

## 1. First:

**Syntax:**

{$first:<expression>}

**Source Code:**

```
use examdb
db.student.aggregate([{$group:{_id:"$Class",Score:{$first:"$Score"}}}])
```

**Output:**

```
> db.student.aggregate([{$group:{_id:"$Class",Score:{$first:"$Score"}}}])
{ "_id" : "C3", "Score" : 90 }
{ "_id" : "C1", "Score" : 95 }
{ "_id" : "C2", "Score" : 75 }
{ "_id" : "Biology", "Score" : 90 }
{ "_id" : "Chemistry", "Score" : 90 }
>
```

## 2. Last:

**Syntax:**

{$last:<expression>}

**Source Code:**

```
use examdb
db.student.aggregate([{$group:{_id:"$Class",Score:{$last:"$Score"}}}])
```

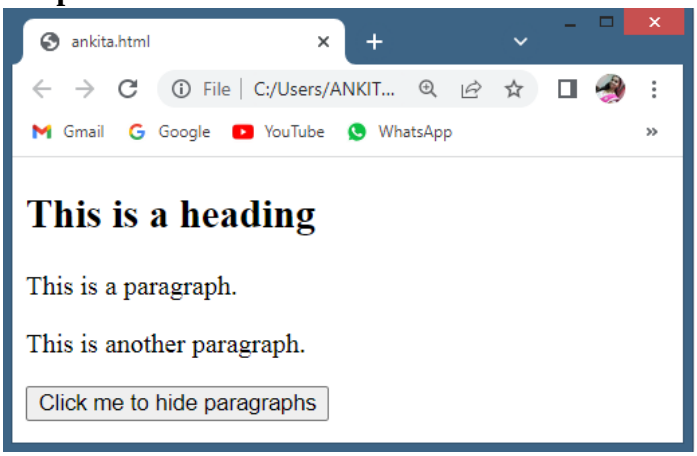
**Output:**

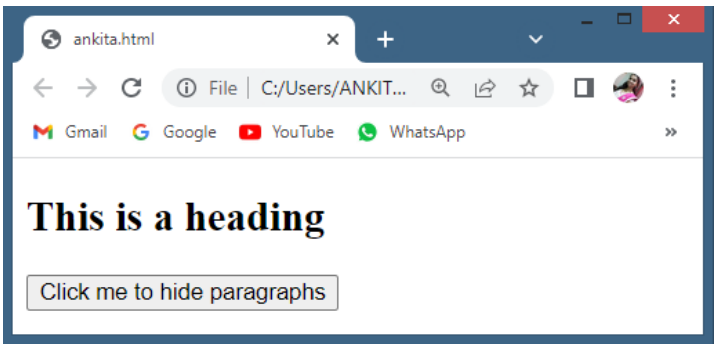
```
> db.student.aggregate([{$group:{_id:"$Class",Score:{$last:"$Score"}}}])
{ "_id" : "C3", "Score" : 90 }
{ "_id" : "C1", "Score" : 75 }
{ "_id" : "C2", "Score" : 90 }
{ "_id" : "Biology", "Score" : 90 }
{ "_id" : "Chemistry", "Score" : 90 }
>
```

**Practical No.: 3**
**Aim: Programs on Basic jQuery**
**a. jQuery Basic, jQuery Events**
**jQuery Basic:**
**Code:**
**index.html**

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function(){
      $("button").click(function(){
            $("p").hide();
      });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me to hide paragraphs</button>
</body>
</html>
```

**Output:**



## jQuery Events:

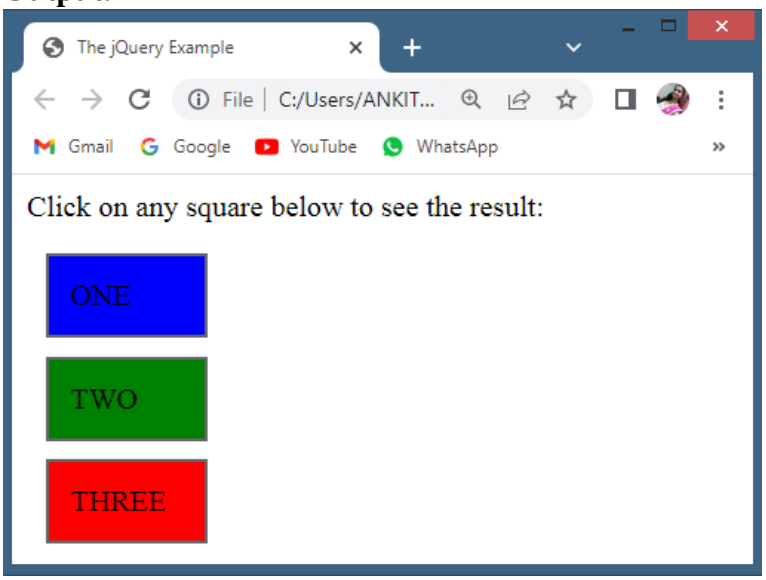
**1) Click Event:**

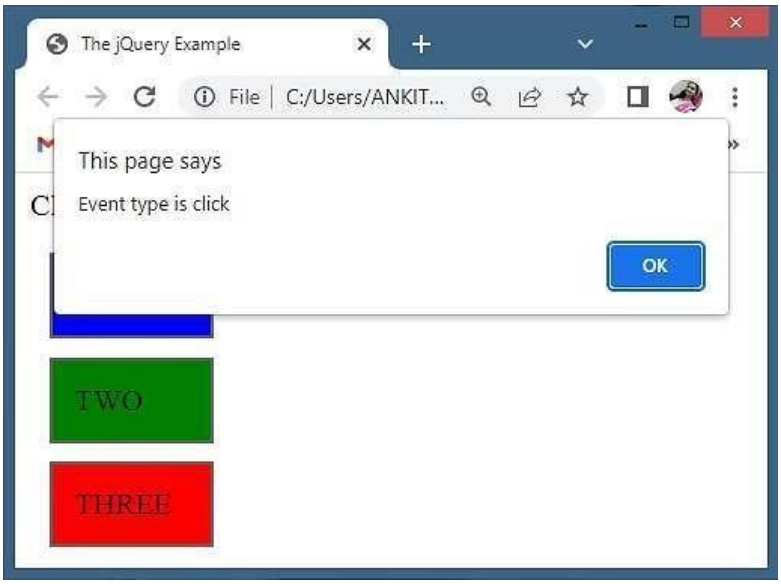
**Code:**

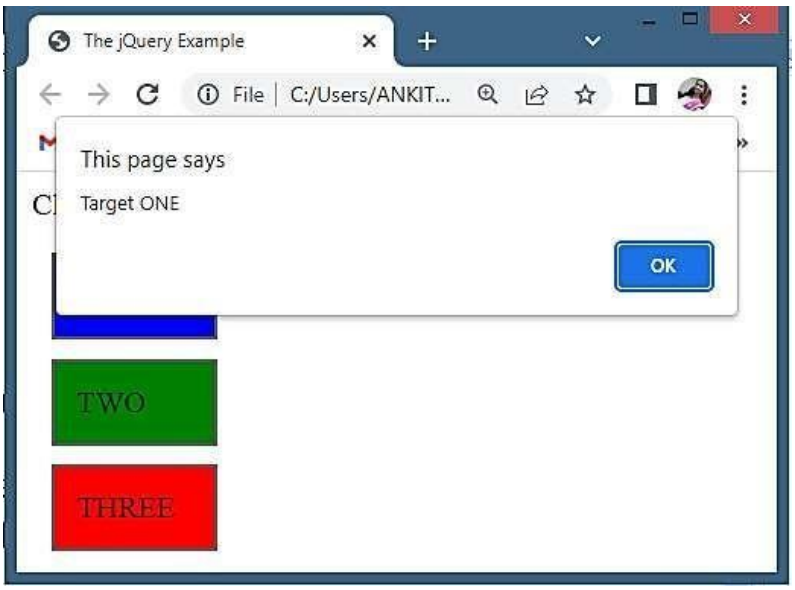
**index.html**

```
<html>
<head>
<title>The jQuery Example</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      $("div").bind("click", function(event) {
            alert("Event type is " + event.type);
            alert("Target " + event.target.innerHTML);
      });
});
</script>
<style>
      .div{margin:10px;padding:12px; border:2px solid #666; width:60px;}
</style>
</head>
<body>

<p>Click on any square below to see the result:</p>

<div class = "div" style = "background-color:blue;">ONE</div>
<div class = "div" style = "background-color:green;">TWO</div>
<div class = "div" style = "background-color:red;">THREE</div>
</body>
</html>
```

**Output:**






**2. DoubleClick Event**

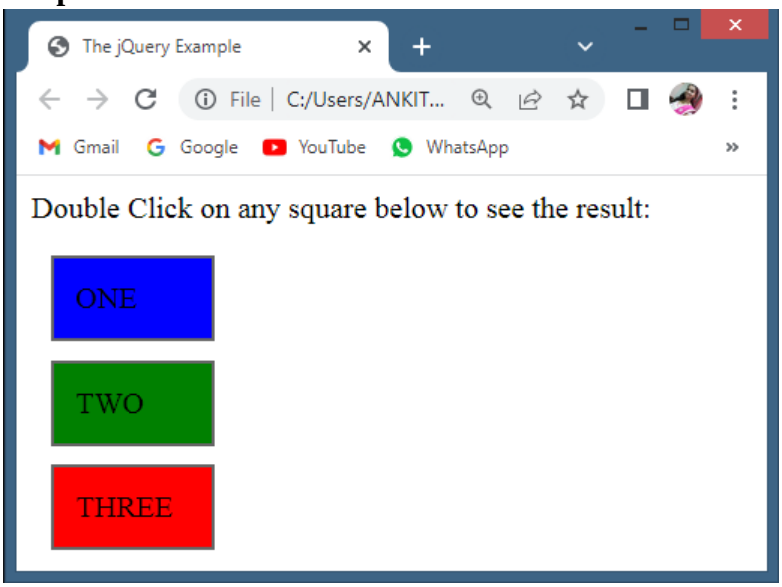
**Code:**

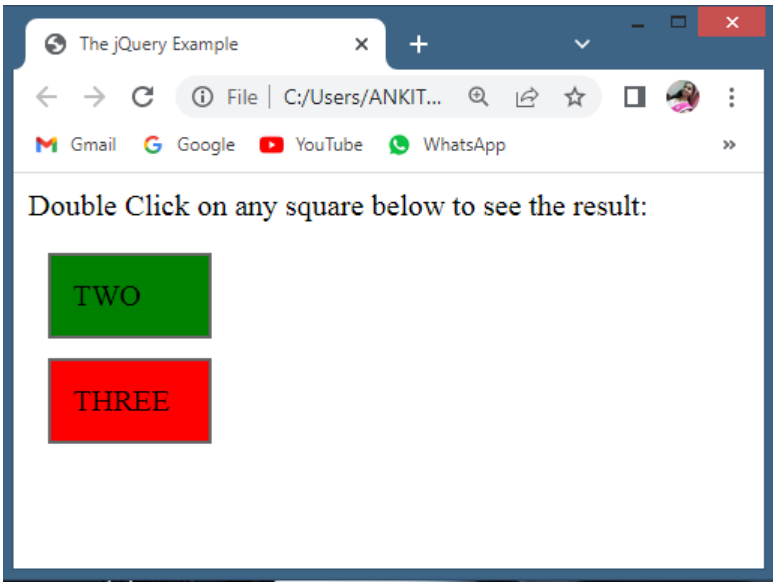
**index.html**

```
<html>
<head>
<title>The jQuery Example</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      $("div").dblclick(function(){
            $(this).hide();
      });
});
</script>
<style>
      .div{margin:10px;padding:12px; border:2px solid #666; width:60px;}
</style>
```

```
</head>
<body>

<p>Double Click on any square below to see the result:</p>

<div class = "div" style = "background-color:blue;">ONE</div>
<div class = "div" style = "background-color:green;">TWO</div>
<div class = "div" style = "background-color:red;">THREE</div>
</body>
</html>
```

**Output:**




**3. Mouseleave Event:**

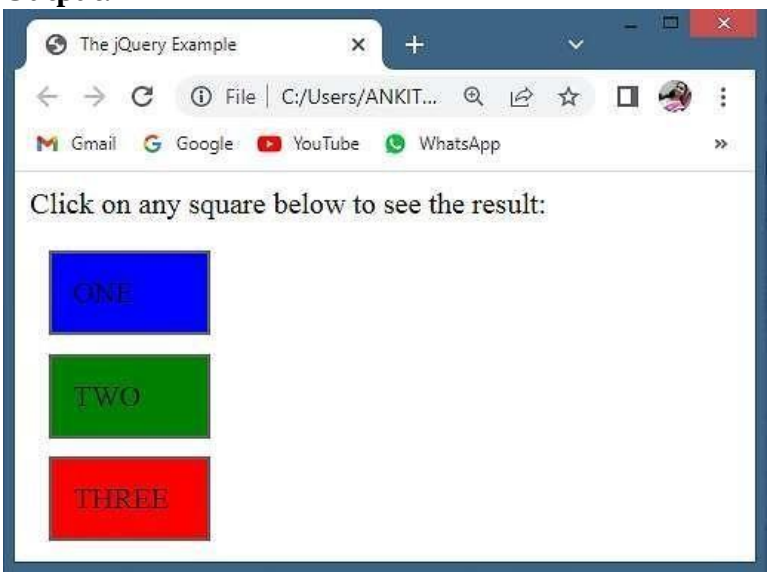
**Code:**

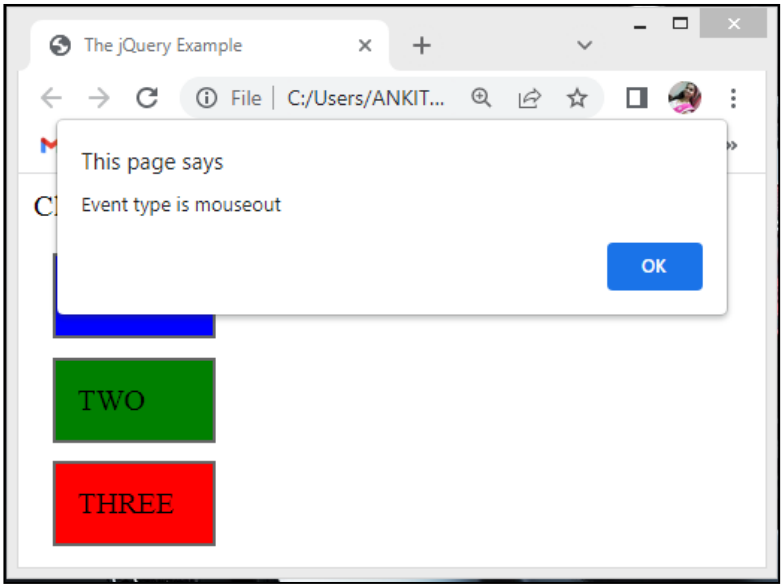
**index.html**

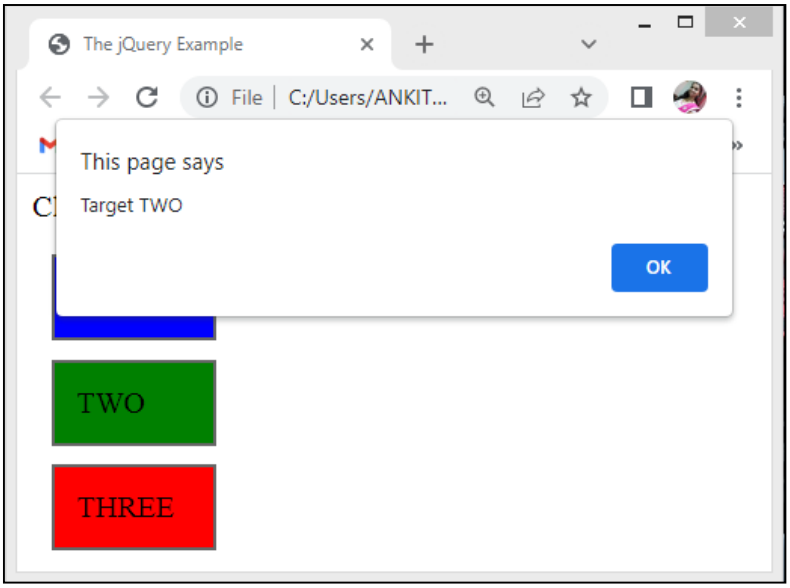
```
<html>
<head>
<title>The jQuery Example</title>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      $(".div").mouseleave(function(){
            alert("Event type is " + event.type);
            alert("Target " + event.target.innerHTML);
      });
});
</script>
<style>
      .div{margin:10px;padding:12px; border:2px solid #666; width:60px;}
</style>
</head>
<body>

<p>Click on any square below to see the result:</p>

<div class = "div" style = "background-color:blue;">ONE</div>
<div class = "div" style = "background-color:green;">TWO</div>
<div class = "div" style = "background-color:red;">THREE</div>
</body>
</html>
```

**Output:**




The jQuery Example
File | C:/Users/ANKIT...
This page says
Event type is mouseout
OK
TWO
THREE


The jQuery Example
File | C:/Users/ANKIT...
This page says
Target TWO
OK
TWO
THREE

## b. jQuery Selectors, jQuery Hide and Show effects

## jQuery Selectors:

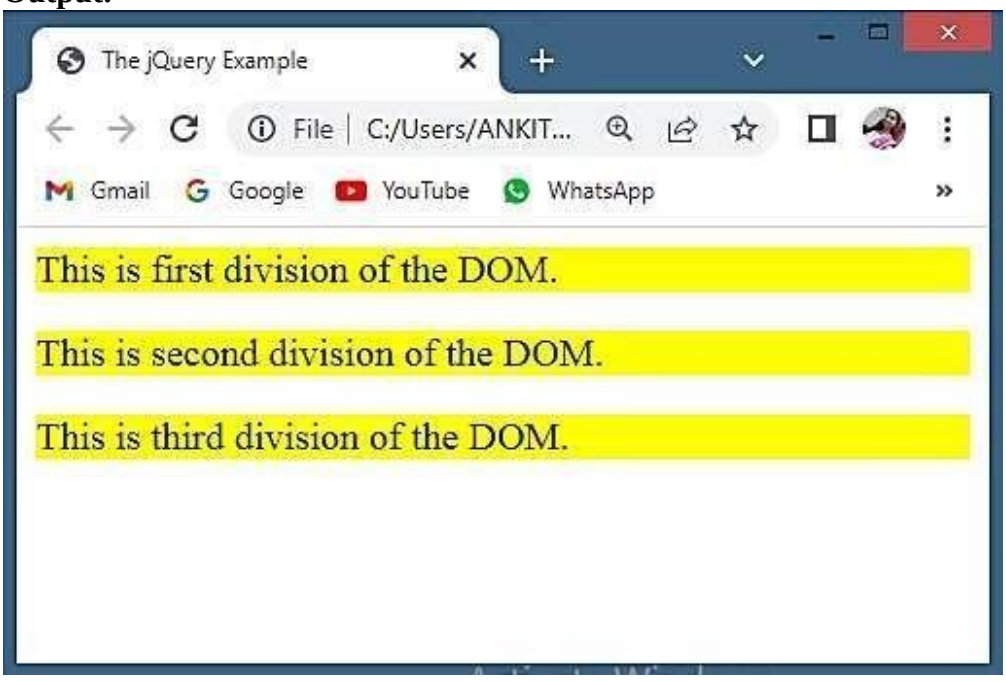
**1. Name Selector**

**Code:**

**index.html**

```
<html>
<head>
<title>The jQuery Example</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      /* This would select all the divisions */
      $("div").css("background-color","yellow");
});
</script>
</head>
<body>
<div class = "big" id="div1">
<p>This is first division of the DOM.</p>
</div>
<div class = "medium" id="div2">
<p>This is second division of the DOM.</p>
</div>
<div class = "small" id="div3">
<p>This is third division of the DOM.</p>
</div>
</body>
</html>
```

**Output:**



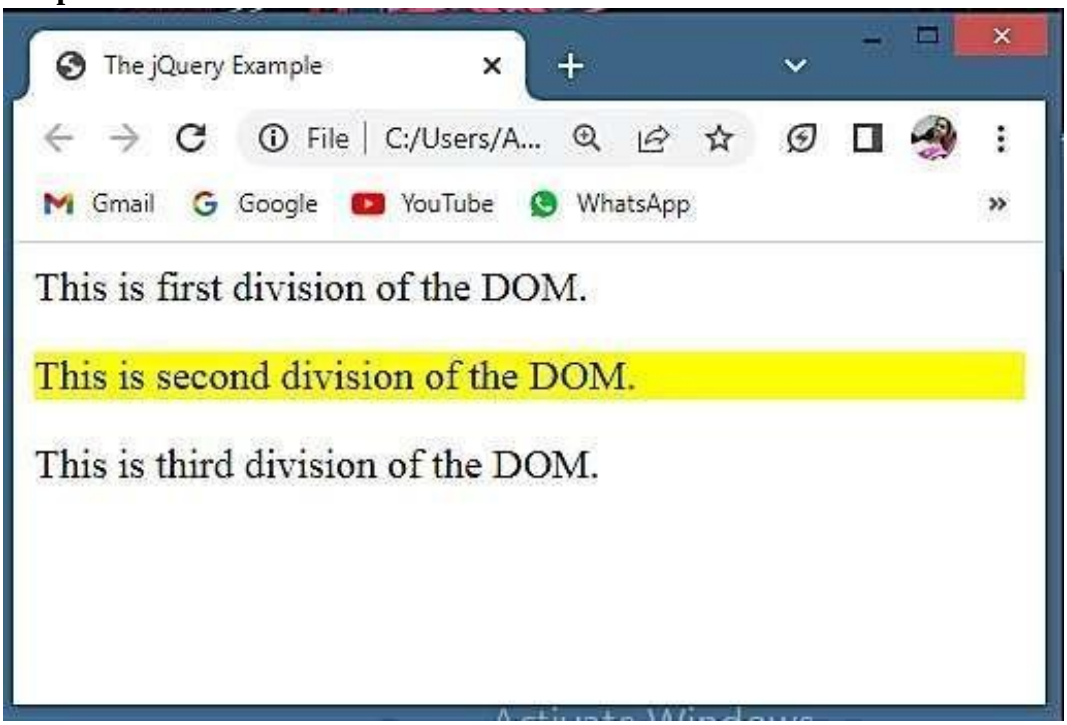
**2. ID Selector**

**Code:**

**index.html**

```
<html>
<head>
<title>The jQuery Example</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      /* This would select 2nd division by id */
      $("#div2").css("background-color","yellow");
});
</script>
</head>
<body>
<div class = "big" id="div1">
<p>This is first division of the DOM.</p>
</div>
<div class = "medium" id="div2">
<p>This is second division of the DOM.</p>
</div>
<div class = "small" id="div3">
<p>This is third division of the DOM.</p>
</div>
</body>
</html>
```

**Output:**



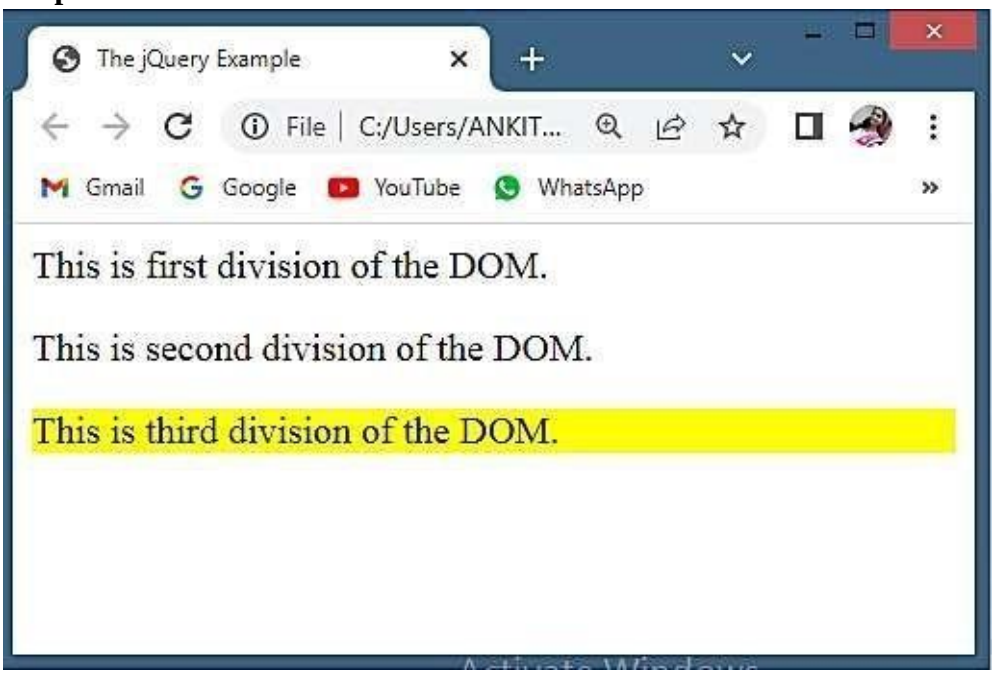
**3. Class Selector**

**Code:**

**index.html**

```
<html>
<head>
<title>The jQuery Example</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      /* This would select 3rd division by class name */
      $(".small").css("background-color","yellow");
});
</script>
</head>
<body>
<div class = "big" id="div1">
<p>This is first division of the DOM.</p>
</div>
<div class = "medium" id="div2">
<p>This is second division of the DOM.</p>
</div>
<div class = "small" id="div3">
<p>This is third division of the DOM.</p>
</div>
</body>
</html>
```

**Output:**



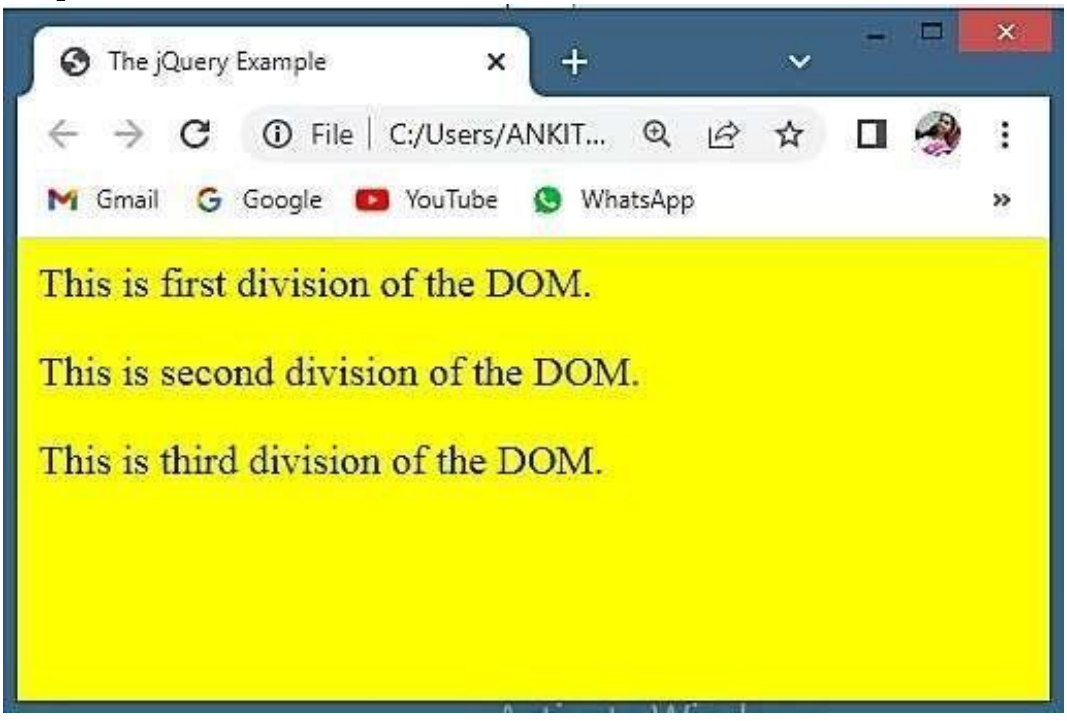
**4. Universal Selector**

**Code:**

**index.html**

```
<html>
<head>
<title>The jQuery Example</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      /* This would select all the divisions */
      $("*").css("background-color","yellow");
});
</script>
</head>
<body>
<div class = "big" id="div1">
<p>This is first division of the DOM.</p>
</div>
<div class = "medium" id="div2">
<p>This is second division of the DOM.</p>
</div>
<div class = "small" id="div3">
<p>This is third division of the DOM.</p>
</div>
</body>
</html>
```

**Output:**



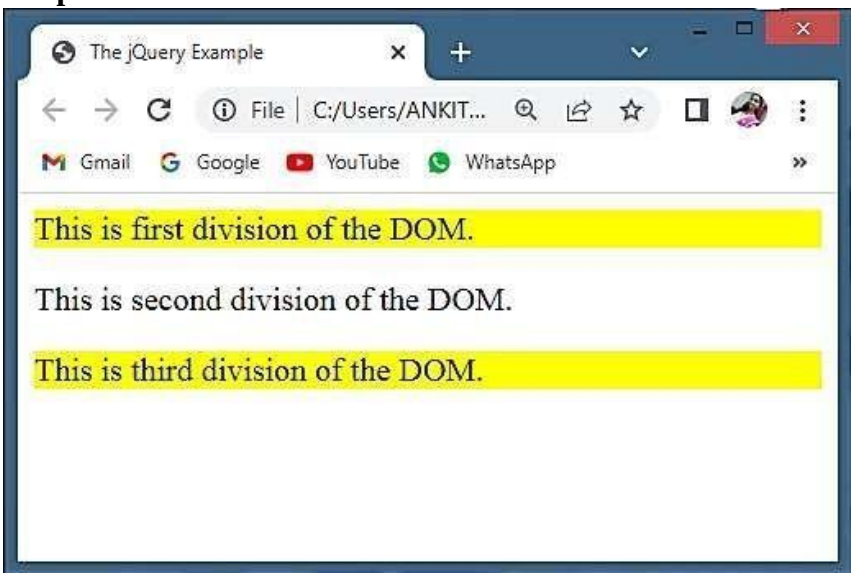
**5. Multiple Selector:**

**Code:**

**index.html:**

```
<html>
<head>
<title>The jQuery Example</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      /* This would select multiple divisions */
      $(".small, #div1").css("background-color","yellow");
});
</script>
</head>
<body>
<div class = "big" id="div1">
<p>This is first division of the DOM.</p>
</div>
<div class = "medium" id="div2">
<p>This is second division of the DOM.</p>
</div>
<div class = "small" id="div3">
<p>This is third division of the DOM.</p>
</div>
</body>
</html>
```

**Output:**



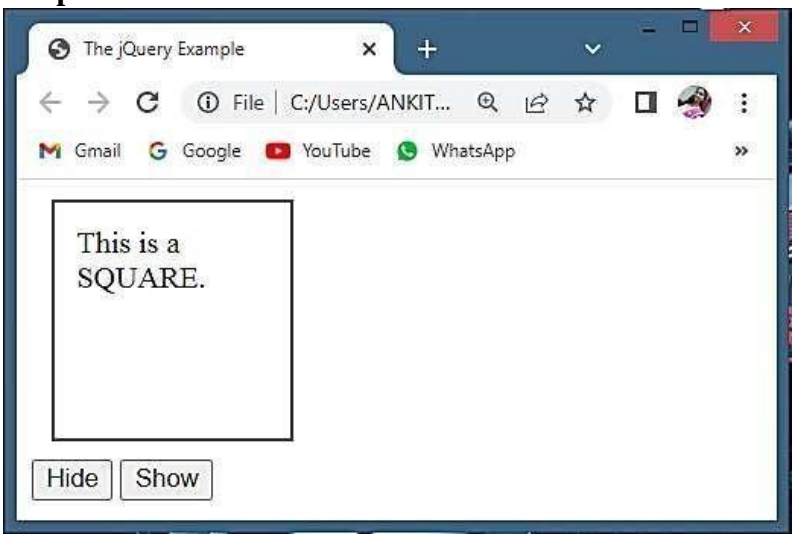
## jQuery Hide and Show Effects:

**Code:**

**index.html**

```
<html>
<head>
<title>The jQuery Example</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      $("#show").click(function(){
            $(".mydiv").show(1000);
      });
      $("#hide").click(function(){
            $(".mydiv").hide(1000);
});
});

</script>
<style>
.mydiv{    margin:10px; padding:12px; border:2px solid #666; width:100px;
height:100px; }
</style>
</head>
<body>
<div class = "mydiv"> This is a SQUARE. </div>
<input id = "hide" type = "button" value = "Hide" />
<input id = "show" type = "button" value = "Show" />
</body>
</html>
```

**Output:**




## c. jQuery fading effects, jQuery Sliding effects

### jQuery Fading Effects:

**Code:**

**index.html**

```
<html>
<head>
<title>The jQuery Example</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      $("#fadeout").click(function(){
            $("h1").fadeOut(2000, function () {
                  alert("Animation is completed");
            });
      });
      $("#fadein").click(function(){
            $("h1").fadeIn(2000, function () {
                  alert("Animation is completed");
            });
      });
});

</script>
</head>
<body align = "center">
<div>
<h1>The Text is Visible. Click Fade Out Button to Hide Text.</h1>
<input id = "fadeout" type = "button" value = "Fade Out" />
<input id = "fadein" type = "button" value = "Fade In" />
</div>
</body>
</html>
```

**Output:**



## jQuery Sliding Effects:

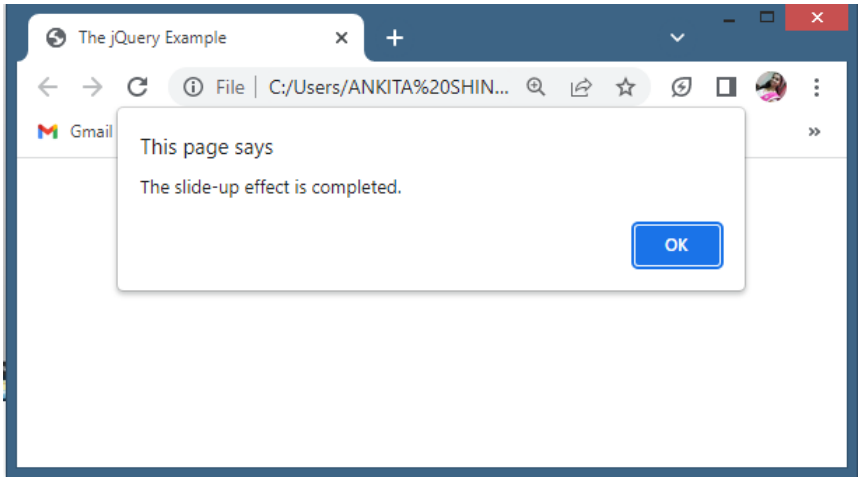
**Code:**

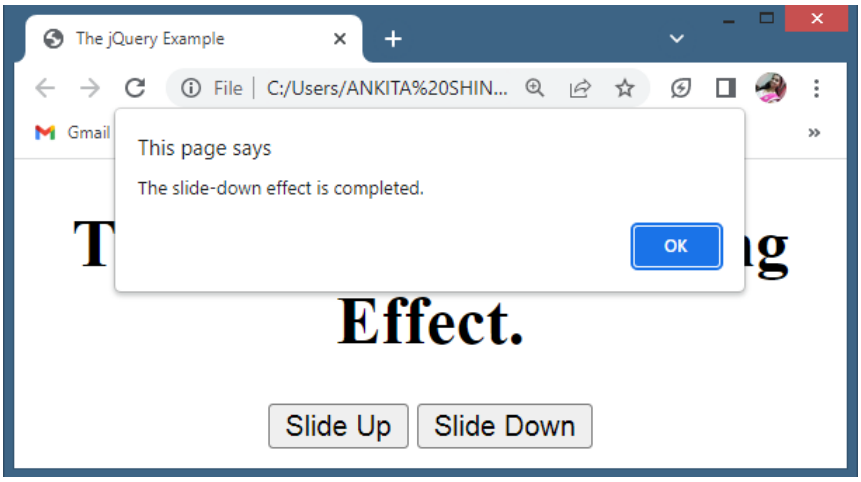
**index.html**

```
<html>
<head>
<title>The jQuery Example</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      $("#slideup").click(function(){
            $("h1").slideUp("slow", function(){
                  alert("The slide-up effect is completed.");
            });
      });
      $("#slidedown").click(function(){
            $("h1").slideDown("slow", function(){
                  alert("The slide-down effect is completed.");
            });
      });
});
```

```
</script>
</head>
<body align = "center">
<div>
<h1>This is a Demo of Sliding Effect.</h1>
<input id = "slideup" type = "button" value = "Slide Up" />
<input id = "slidedown" type = "button" value = "Slide Down" />
</div>
</body>
</html>
```

**Output:**


The jQuery Example
File | C:/Users/ANKITA%20SHIN...
Gmail
This page says
The slide-down effect is completed.
OK
Effect.
Slide Up
Slide Down

**Practical No.: 4**

## Aim: jQuery Advance

## jQuery Chaining:

**Code:**

**index.html**

```
<html>
<head>
<title>The jQuery Example</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function() {
      $("#chain").click(function(){
            $("h1").css("color" , "red").slideUp(2000).slideDown(2000);
      });
});
</script>
</head>
<body>
<h1>A simple Chaining example:</h1>
<input id = "chain" type = "button" value = "Click Me...!!!" />
</body>
</html>
```

**Output:**


The jQuery Example
File | C:/Users/ANKITA%20...
Gmail
Google
YouTube
WhatsApp
A simple Chaining example:
Click Me...!!!


The jQuery Example
File | C:/Users/ANKITA%20...
Gmail
Google
YouTube
WhatsApp
A simple Chaining example:
Click Me...!!!

## Practical No.:5

## Aim: JSON

## a. Creating JSON

**Code:**

**index.html**

```
<html>
<head>
<title>The JSON Example</title>
</head>
<body>

<p>Access a JSON object using dot notation:</p>
<p id = "demo"></p>
<script>
var myObj, x;
myObj = {
            "name":"John",
            "age":30,
            "car":null
        };
x = myObj.name;
document.getElementById("demo").innerHTML = x;
</script>

</body>
</html>
```

**Output:**

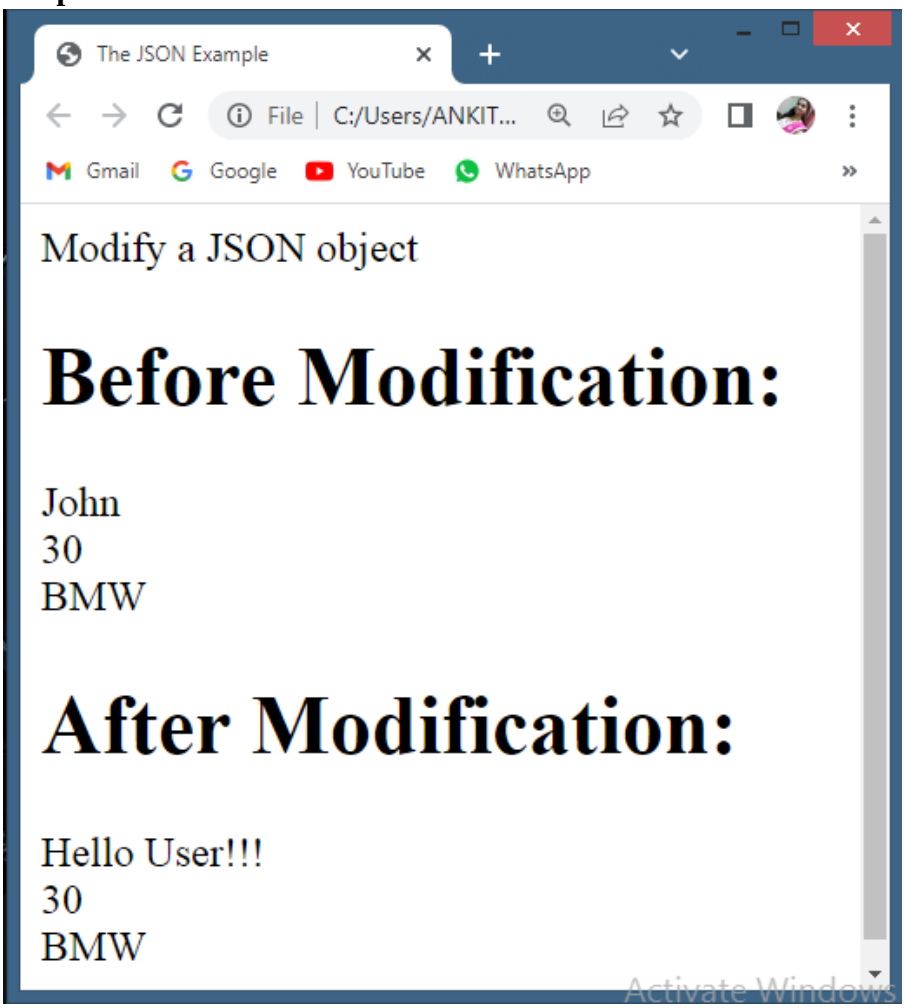
## b. Parsing JSON

**Code:**

**index.html**

```
<html>
<head>
<title>The JSON Example</title>
</head>
<body>
<p>Modify a JSON object</p>

<h1>Before Modification:</h1>
<p id = "demo1"></p>

<h1>After Modification:</h1>
<p id = "demo2"></p>

<script>
var myObj, x = "",i;
myObj = {
           "name":"John",
           "age":30,
           "car":"BMW"
        };
for(i in myObj)
{
      x = x + myObj[i] + "<br>";
}
document.getElementById("demo1").innerHTML = x;

x = "";
myObj.name = "Hello User!!!";

for(i in myObj)
{
      x = x + myObj[i] + "<br>";
}
document.getElementById("demo2").innerHTML = x;
</script>
</body>
</html>
```

**Output:**


The JSON Example
File | C:/Users/ANKIT...
Gmail
Google
YouTube
WhatsApp
Modify a JSON object
Before Modification:
John
30
BMW
After Modification:
Hello User!!!
30
BMW
Activate Windows

**c. Persing JSON**

**Code:**

**index.html**

```
<html>
<head>
<title>The JSON Example</title>
</head>
<body>

<p>Create Object from JSON String</p>
<p id = "demo"></p>
<script>
var txt = '{"name":"John" , "age":30 , "car":"BMW"}'

var obj = JSON.parse(txt);

document.getElementById("demo").innerHTML = obj.name + " , " + obj.age + " ,
" + obj.car;
</script>

</body>
</html>
```

**Output:**

**Practical No.: 6**

**Aim: MongoDB and JSON**

**a. Create a JSON file and import it to MongoDB**

**Steps:**

1. Open a Text editor like Notepad, Visual Studio Code.
2. Type following data:

```
[
  {
    "id": 1,
    "name": "John Deo",
    "class": "Four",
    "mark": 75,
    "gender": "female"
  },
  {
    "id": 2,
    "name": "Max Ruin",
    "class": "Three",
    "mark": 85,
    "gender": "male"
  },
  {
    "id": 3,
    "name": "Arnold",
    "class": "Three",
    "mark": 55,
    "gender": "male"
  },
  {
    "id": 4,
    "name": "Krish Star",
    "class": "Four",
    "mark": 60,
    "gender": "female"
  },
  {
    "id": 5,
    "name": "John Mike",
    "class": "Four",
    "mark": 60,
    "gender": "female"
  },
  {
    "id": 6,
    "name": "Alex John",
```

```
    "class": "Four",
    "mark": 55,
    "gender": "male"
  },
  {
    "id": 7,
    "name": "My John Rob",
    "class": "Fifth",
    "mark": 78,
    "gender": "male"
  },
  {
    "id": 8,
    "name": "Asruid",
    "class": "Five",
    "mark": 85,
    "gender": "male"
  },
  {
    "id": 9,
    "name": "Tes Qry",
    "class": "Six",
    "mark": 78,
    "gender": "male"
  },
  {
    "id": 10,
    "name": "Big John",
    "class": "Four",
    "mark": 55,
    "gender": "female"
  }
]
```

3. Save the file with the extension of .json. Here we have to give the name to the file as student.json
4. Now open Windows Command Prompt (cmd) and type following command.
   Note: Search cmd then right click on that and select option 'Run as Administrator'.

```
> cd\
> cd "C:\Program Files\MongoDB\Server\4.2\bin"
> mongoimport --db studentDB --collection student --file student.json --jsonArray
```

   After that it will give output that your all documents imported successfully.

5. Open MongoDB Console. Execute following commands:

```
> show dbs
> use studntDB
> show collections
> db.student.find()
```

**Output:**

```
> show dbs
StudentDB  0.000GB
admin      0.000GB
config     0.000GB
examdb     0.000GB
local      0.000GB
>
> use StudentDB
switched to db StudentDB
>
> show collections
student
>
> db.student.find()
{ "_id" : 1, "name" : "John Deo", "class" : "Four", "mark" : 75, "gender" : "female" }
{ "_id" : 2, "name" : "Max Ruin", "class" : "Three", "mark" : 85, "gender" : "male" }
{ "_id" : 3, "name" : "Arnold", "class" : "Three", "mark" : 55, "gender" : "male" }
{ "_id" : 5, "name" : "John Mike", "class" : "Four", "mark" : 60, "gender" : "female" }
{ "_id" : 6, "name" : "Alex John", "class" : "Four", "mark" : 55, "gender" : "male" }
{ "_id" : 4, "name" : "Krish Star", "class" : "Four", "mark" : 60, "gender" : "female" }
{ "_id" : 7, "name" : "My John Rob", "class" : "Fifth", "mark" : 78, "gender" : "male" }
{ "_id" : 10, "name" : "Big John", "class" : "Four", "mark" : 55, "gender" : "female" }
{ "_id" : 8, "name" : "Asruid", "class" : "Five", "mark" : 85, "gender" : "male" }
{ "_id" : 9, "name" : "Tes Qry", "class" : "Six", "mark" : 78, "gender" : "male" }
>
>
```

## b. Export MongoDB to JSON.

**Reference Link -**
https://www.nielit.gov.in/gorakhpur/sites/default/files/Gorakhpur/ALEVEL_1_DBTECH_08_June_2020_IL.pdf

**Steps:**

1. Open Windows Command Prompt (cmd) and type following command.
   Note: Search cmd then right click on that and select option 'Run as Administrator'.

```
> cd\
> cd " C:\Program Files\MongoDB\Server\4.2\bin"
> mongoexport --db StudentDB --collection student --type=json --out
E:\sampleResult.json
```

   After that it will give output that your all documents imported successfully.

2. You can check your JSON file as you specified directory in above command.

## Output: