Sahyog College of Management Studies, Thane (W)

Affiliated to Mumbai University

Course : BSC (Information Technology)
Semester : V

Subject : Emerging Technology

Lab Manual

Practical No : 1

a) **Write a MongoDB query to create and drop database.**
b) **Write a MongoDB query to create, display and drop collection**
c) **Write a MongoDB query to insert, query, update and delete a document**

a. Write a MongoDB query to create and drop database
Solution :
Creation of database: use college;
Deletion of database: db.dropDatabase()

b. Write a MongoDB query to create, display and drop collection
Solution :
Creation a Collection: db.createCollection("Student")
Display a Collection: db.Student.find()
Drop a Collection : db.Student.drop()

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

<u>Insertion of Sinlge Document :</u>

```
db.Student.insertOne(
  {
          First_Name: "Radhika",
          Last_Name: "Sharma",
          Date_Of_Birth: "1995-09-26",
          e_mail: "radhika_sharma.123@gmail.com",
          phone: "9848022338"
  })
```

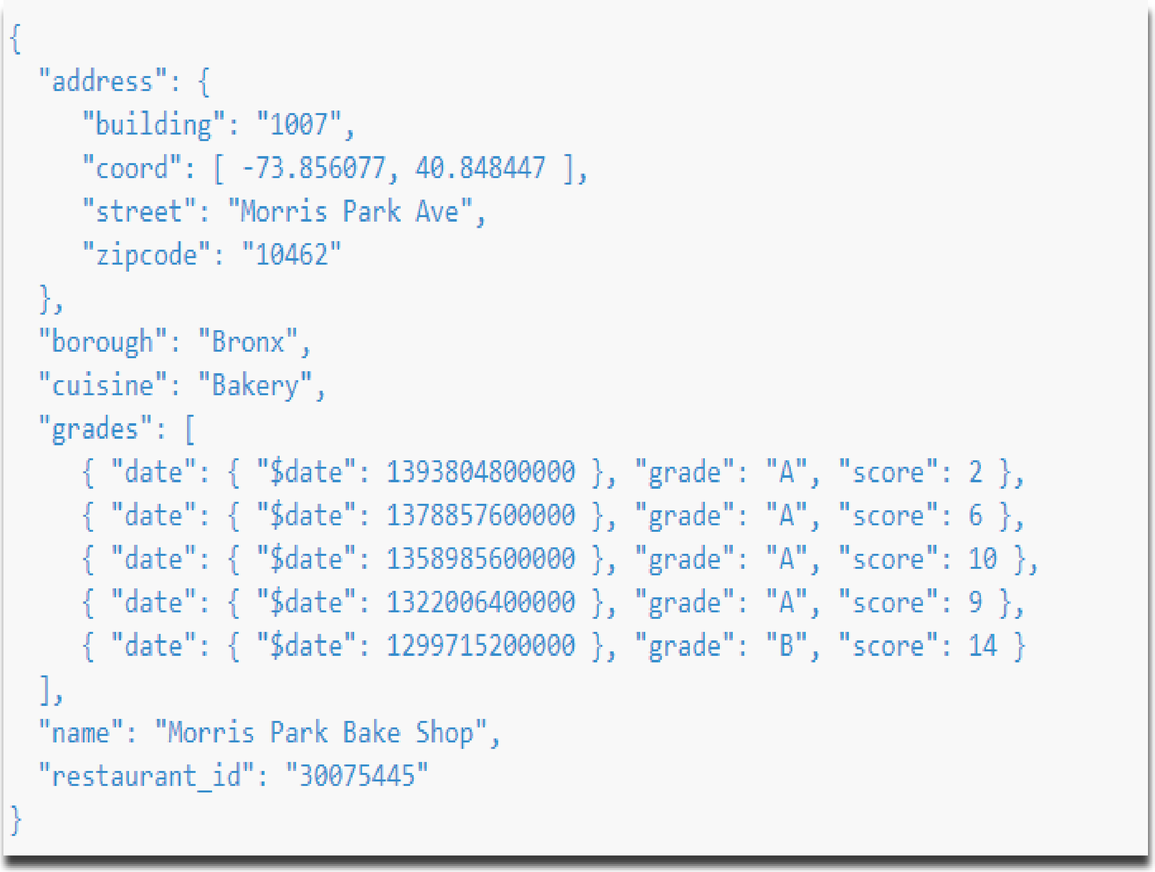
<u>Insertion of Multiple Document :</u>

```
db.empDetails.insertMany(
  [
          {
                First_Name: "Radhika",
                Last_Name: "Sharma",
                Date_Of_Birth: "1995-09-26",
                e_mail: "radhika_sharma.123@gmail.com",
                phone: "9000012345"
        },
        {
                First_Name: "Rachel",
                Last_Name: "Christopher",
                Date_Of_Birth: "1990-02-16",
                e_mail: "Rachel_Christopher.123@gmail.com",
                phone: "9000054321"
        },
        {
                First_Name: "Fathima",
                Last_Name: "Sheik",
                Date_Of_Birth: "1990-02-16",
                e_mail: "Fathima_Sheik.123@gmail.com",
                phone: "9000054321"
        }
  ]
)
```

Practical No : 2

a) Simple Queries with MongoDB

Sample Data :



```
{
  "address": {
     "building": "1007",
     "coord": [ -73.856077, 40.848447 ],
     "street": "Morris Park Ave",
     "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
     { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
     { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
     { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
     { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
     { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```

1. Write a MongoDB query to display all the documents in the collection restaurants.

Solution :

db.restaurants.find();

2. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine for all the documents in the collection restaurant.

Solution :

db.restaurants.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1});

3. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine, but exclude the field _id for all the documents in the collection restaurant.

Solution :

```
db.restaurants.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine"
:1,"_id":0});
```

4. Write a MongoDB query to display all the restaurant which is in the borough Bronx

Solution :

```
db.restaurants.find({"borough": "Bronx"});
```

5. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.

Solution :

```
db.restaurants.find({grades : { $elemMatch:{"score":{$gt : 80 , $lt :100}}}});
```

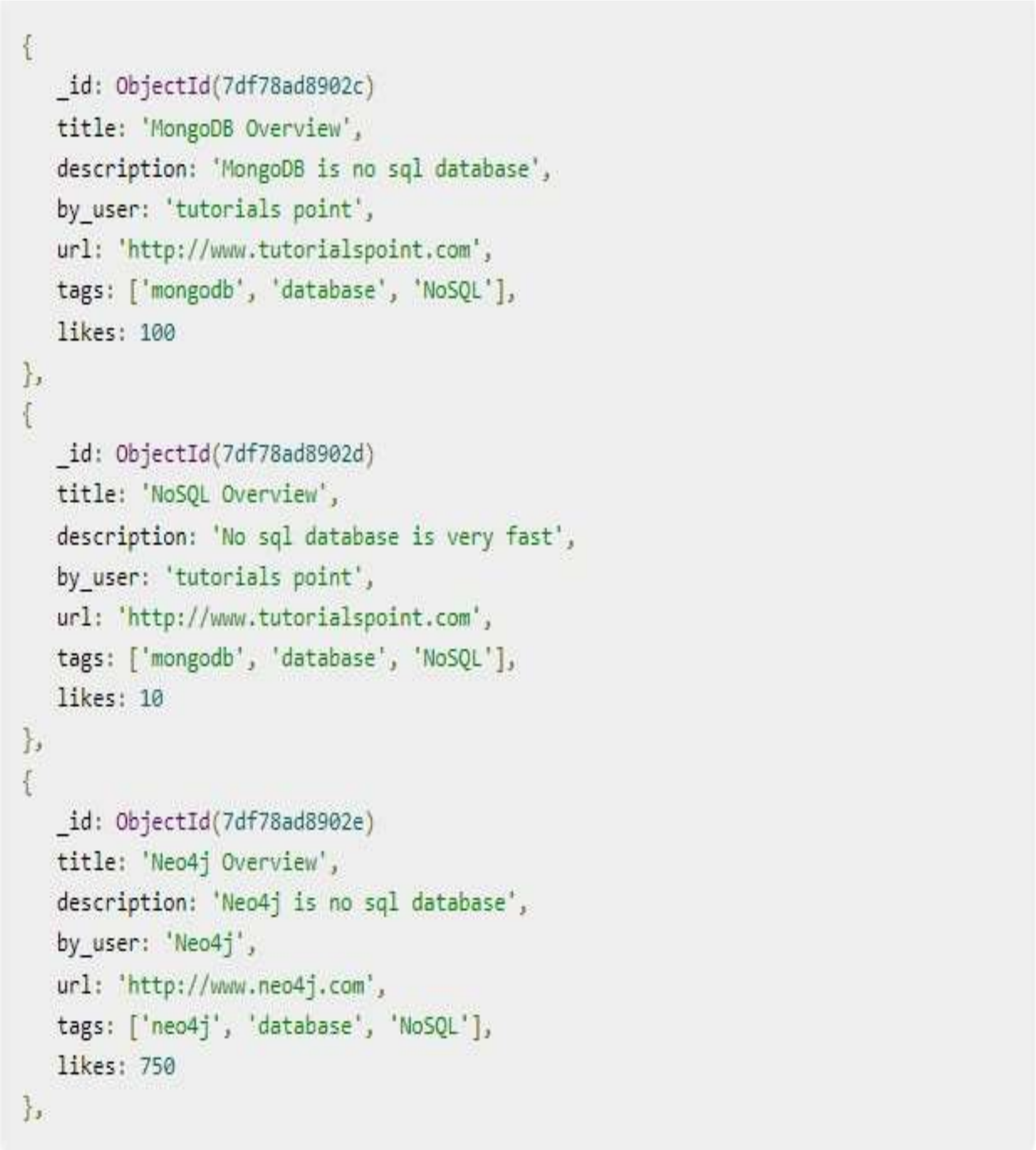
Practical No : 3

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

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

c) Write a MongoDB query to use first and last expression

Sample Data :



```
{
   _id: ObjectId(7df78ad8902c)
   title: 'MongoDB Overview',
   description: 'MongoDB is no sql database',
   by_user: 'tutorials point',
   url: 'http://www.tutorialspoint.com',
   tags: ['mongodb', 'database', 'NoSQL'],
   likes: 100
},
{
   _id: ObjectId(7df78ad8902d)
   title: 'NoSQL Overview',
   description: 'No sql database is very fast',
   by_user: 'tutorials point',
   url: 'http://www.tutorialspoint.com',
   tags: ['mongodb', 'database', 'NoSQL'],
   likes: 10
},
{
   _id: ObjectId(7df78ad8902e)
   title: 'Neo4j Overview',
   description: 'Neo4j is no sql database',
   by_user: 'Neo4j',
   url: 'http://www.neo4j.com',
   tags: ['neo4j', 'database', 'NoSQL'],
   likes: 750
},
```

a) Write a MongoDB Query to use sum , avg , min and max expression.

Solution :

- db.User.aggregate([{$group : {_id : "$by_user", num_tutorial : {$sum : "$likes"}}}])

- db.User.aggregate([{$group : {_id : "$by_user", num_tutorial : {$avg : "$likes"}}}])

- db.User.aggregate([{$group : {_id : "$by_user", num_tutorial : {$min : "$likes"}}}])

- db.User.aggregate([{$group : {_id : "$by_user", num_tutorial : {$max : "$likes"}}}])

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

Solution :

- db.User.aggregate([{$group : {_id : "$by_user", url : {$push: "$url"}}}])

- db.User.aggregate([{$group : {_id : "$by_user", url : {$addToSet : "$url"}}}])

c) Write a MongoDB query to use first and last expression

Solution :

- db.User.aggregate([{$group : {_id : "$by_user", first_url : {$first : "$url"}}}])

- db.User.aggregate([{$group : {_id : "$by_user", last_url : {$last : "$url"}}}])

Practical No : 4

a) Write a MongoDB query to create Replica of existing database
b) Write a MongoDB query to create a backup of existing database
c) Write a MongoDB query to restore database from the backup.

a) Write a MongoDB query to create Replica of Existing database.

Solution :

```
1.create folder "Data"
2.create 3 sub folders within "Data" : rs1,rs2,rs3
3.open cmd in C:\Program Files\MongoDB\Server\4.4\bin
4."utube" = server name
  start mongod -replSet utube -logpath F:\NGT\Replica\Data\rs1\1.log --
dbpath F:\NGT\Replica\Data\rs1 --port 27018
  start mongod -replSet utube -logpath F:\NGT\Replica\Data\rs2\2.log --
dbpath F:\NGT\Replica\Data\rs2 --port 27019
  start mongod -replSet utube -logpath F:\NGT\Replica\Data\rs3\3.log --
dbpath F:\NGT\Replica\Data\rs3 --port 27020

///Primary==================================================
5.again open cmd in C:\Program Files\MongoDB\Server\4.4\bin
  mongo --port 27018
6.Configure Server
  config={_id:"utube",members:[{_id:0,host:"localhost:27018"},{_id:1,host:"
localhost:27019"},{_id:2,host:"localhost:27020"}]}
7.        rs.initiate(config)
8.        rs.status()
9.shift to primary ( 27018 )
10. create database and collection and insert docs
  use test123
  db.createCollection("cust")
  db.cust.insert({"name":"hardik"})
```

```
///secondary======================================================
11. open cmd in C:\Program Files\MongoDB\Server\4.4\bin
  mongo --port 27019
12. "show dbs" not work in 27019 cmd
13."rs.slaveOk()" -->even this wont work so run
  rs.secondaryOk()
15.access all on secondary but we can't write/modify on secondary (i.e.
27019/27020)
```

b) Write a Write a MongoDB query to create a backup of existing database

Solution :

Backup : mongodump –db Hotel –collection Employee –out c:\backup

c) Write a MongoDB query to restore database from the backup.

Solution :

Restore : mongorestore –db Hotel –collection students c:\Test\ABC\students.bson

Practical No : 5

Connecting Java with MongoDB and inserting, retrieving, updating and deleting

Solution :

Instruction :

1. Open jdk\jre\lib\ext
2. paste java driver file
3. set classpath
4. Type program
5. compile and run the program

```
import com.mongodb.client.FindIterable;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.model.Filters;
import java.util.Iterator;
import org.bson.Document;
import com.mongodb.MongoClient;
import com.mongodb.BasicDBObject;

public class ConnectToDB1
{
  public static void main( String args[] )
  {
    // Creating a Mongo client
    MongoClient mongo = new MongoClient( "localhost" , 27017 );

   // Accessing the database
    MongoDatabase database = mongo.getDatabase("temp");
    System.out.println("Databse connected");

   MongoCollection<Document> collection =
database.getCollection("test1"); //accessing collection
```

```
      System.out.println("Collection sampleCollection1 selected
successfully");

     // Inserting Documents

      Document document = new Document();
      document.append("name", "B");
      document.append("age", 15);

      collection.insertOne(document);
     System.out.println("Document inserted successfully");

       // Deleting the Documents
      collection.deleteMany(Filters.eq("name", "B"));
      System.out.println("Document deleted successfully...");

      // updating documents
     BasicDBObject updateDocument = new BasicDBObject();
     updateDocument.append("$set", new BasicDBObject().append("name",
"C"));

     BasicDBObject searchQuery2 = new BasicDBObject().append("name",
"A");

     collection.updateMany(searchQuery2, updateDocument);

     System.out.println("Document Updated successfully...");

     FindIterable<Document> iterDoc = collection.find(); // Getting the
iterable object
      int i = 1;
      Iterator it = iterDoc.iterator(); // Getting the iterator
         while(it.hasNext())
     {
                    System.out.println(it.next());
                    i++;
      }
   }
}
```

Practical No : 7

Connecting PYTHON with MongoDB and inserting, retrieving, updating and deleting

Solution :

Instruction :

1. install mongo
2. install python
3. install pip curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py python get-pip.py
4. Open Cmd
5. C:\Users\SCMS\AppData\Local
   . python -m pip install pymongo==3.11
   . check by pip freeze
6. Open idle
7. import pymongo
8. Open file save into folder

```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["test"]
print("db connected")

mycol = mydb["test1"]
print("collection connected")

#inserted
mydict = { "name": "Sunita", "age": 60 }
x = mycol.insert_one(mydict)
print("inserted")

#updated
```

```
myquery = { "name": "Sunita" }
newvalues = { "$set": { "age": 100 } }
mycol.update_many(myquery, newvalues)
print("updated")

#deleted
myquery = { "name": "Sunita" }
mycol.delete_many(myquery)
print("deleted")

#Select data
for x in mycol.find():
  print(x)
```

Practical No : 8

a) jQuery Basic, jQuery Events
b) jQuery Selectors, jQuery Hide and Show effects
c) jQuery fading effects, jQuery Sliding effects

a) JQuery Basic , JQuery Events

Solution :

<u>JQuery Basic :</u>

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/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>
```

JQuery Events :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

b) jQuery Selectors, jQuery Hide and Show effects
Solution :

JQuery Selector :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
```

```
$(document).ready(function(){
  $("button").click(function(){
    $("#test").hide();
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

JQuery Hide/Show :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("#hide").click(function(){
    $("p").hide();
  });
  $("#show").click(function(){
    $("p").show();
  });
});
</script>
</head>
<body>

<p>If you click on the "Hide" button, I will disappear.</p>

<button id="hide">Hide</button>
```

```
<button id="show">Show</button>

</body>
</html>
```

JQuery FadeIN/FadeOut :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeIn();
    $("#div2").fadeIn("slow");
    $("#div3").fadeIn(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeIn() with different parameters.</p>

<button>Click to fade in boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-
color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-
color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-
color:blue;"></div>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeOut();
    $("#div2").fadeOut("slow");
    $("#div3").fadeOut(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeOut() with different parameters.</p>

<button>Click to fade out boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;background-
color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-
color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-
color:blue;"></div>

</body>
</html>

<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
```

```
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeToggle();
    $("#div2").fadeToggle("slow");
    $("#div3").fadeToggle(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeToggle() with different speed parameters.</p>

<button>Click to fade in/out boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;background-
color:red;"></div>
<br>
<div id="div2" style="width:80px;height:80px;background-
color:green;"></div>
<br>
<div id="div3" style="width:80px;height:80px;background-
color:blue;"></div>

</body>
</html>
```

JQuery Sliding :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown("slow");
  });
});
</script>
<style>
#panel, #flip {
padding: 5px;
  text-align: center;
  background-color: #e5eecc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>

<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>

</body>
</html>

<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideUp("slow");
  });
});
</script>
<style>
```

```
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: #e5eecc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 50px;
}
</style>
</head>
<body>

<div id="flip">Click to slide up panel</div>
<div id="panel">Hello world!</div>

</body>
</html>

<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideToggle("slow");
  });
});
</script>
<style>
#panel, #flip {
padding: 5px;
  text-align: center;
  background-color: #e5eecc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>

<div id="flip">Click to slide the panel down or up</div>
<div id="panel">Hello world!</div>

</body>
</html>
```

Practical No : 9

a) **jQuery Animation effects, jQuery Chaining**
b) **jQuery Callback, jQuery Get and Set Contents**
c) **jQuery Insert Content, jQuery Remove Elements and Attribute**

a) JQuery Animation Effects , JQuery Chaining

Solution :

<u>Animation Effects :</u>

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({left: '250px'});
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be
moved. To manipulate the position, remember to first set the CSS position
property of the element to relative, fixed, or absolute!</p>

<div
style="background:#98bf21;height:100px;width:100px;position:absolute;"
></div>
```

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

<u>JQuery Chaining :</u>

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#p1").css("color", "red").slideUp(2000).slideDown(2000);
  });
});
</script>
</head>
<body>

<p id="p1">jQuery is fun!!</p>

<button>Click me</button>

</body>
</html>
```

b) jQuery Callback, jQuery Get and Set Contents

Solution :

<u>JQuery Callback :</u>

```
<!DOCTYPE html>
<html>
<head>
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide("slow", function(){
      alert("The paragraph is now hidden");
    });
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

</body>
</html>
```

jQuery Get and Set Contents :

get Content :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    alert("Text: " + $("#test").text());
  });
  $("#btn2").click(function(){
    alert("HTML: " + $("#test").html());
  });
});
</script>
</head>
<body>

<p id="test">This is some <b>bold</b> text in a paragraph.</p>

<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>

</body>
</html>
```

JQuery Set Content :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text("Hello world!");
  });
  $("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");
  });
  $("#btn3").click(function(){
    $("#test3").val("Dolly Duck");
  });
});
</script>
</head>
<body>

<p id="test1">This is a paragraph.</p>
```

```
<p id="test2">This is another paragraph.</p>

<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>

<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>

</body>
</html>
```

c) jQuery Insert Content, jQuery Remove Elements and Attribute

Solution :

JQuery Insert Element :

Append() :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").append(" <b>Appended text</b>.");
  });

  $("#btn2").click(function(){
    $("ol").append("<li>Appended item</li>");
  });
});
</script>
</head>
```

```
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>

<button id="btn1">Append text</button>
<button id="btn2">Append list items</button>

</body>
</html>
```

Prepend() :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").prepend("<b>Prepended text</b>. ");
  });
  $("#btn2").click(function(){
    $("ol").prepend("<li>Prepended item</li>");
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>

<button id="btn1">Prepend text</button>
<button id="btn2">Prepend list item</button>

</body>
</html>
```

Insert Before and After :

```
<!DOCTYPE html>
<html>
<head>
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>

<script>

$(document).ready(function(){

 $("#btn1").click(function(){

  $("img").before("<b>Before</b>");

 });


 $("#btn2").click(function(){

  $("img").after("<i>After</i>");

 });

});

</script>

</head>

<body>


<img src="/images/w3jquery.gif" alt="jQuery" width="100"
height="140"><br><br>


<button id="btn1">Insert before</button>

<button id="btn2">Insert after</button>


</body>

</html>
```

JQuery Remove Element :

Remove() :

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.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:yellow;">

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>
```

Empty :

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").empty();
  });
});
</script>
</head>
<body>
```

```
<div id="div1" style="height:100px;width:300px;border:1px solid
black;background-color:yellow;">


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>Empty the div element</button>


</body>

</html>
```

Set Attribute :

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
/script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#w3s").attr("href", "https://www.w3schools.com/jquery/");
  });
});
</script>
</head>
<body>
```

```
<p><a href="https://www.w3schools.com"
id="w3s">W3Schools.com</a></p>

<button>Change href Value</button>

<p>Mouse over the link (or click on it) to see that the value of the href
attribute has changed.</p>

</body>
</html>
```

Practical No : 10

a) Creating JSON
b) Parsing JSON
c) Persisting JSON

a) Creating JSON

Solution :

```
<!DOCTYPE html>
<html>
<body>

<h2>Create Object from JSON String</h2>

<p id="demo"></p>

<script>
let text = '{"employees":[' +
'{"firstName":"John","lastName":"Doe" },' +
'{"firstName":"Anna","lastName":"Smith" },' +
'{"firstName":"Peter","lastName":"Jones" }]}';

const obj = JSON.parse(text);
document.getElementById("demo").innerHTML =
obj.employees[1].firstName + " " + obj.employees[1].lastName;
</script>

</body>
</html>
```

b) Parsing JSON

Solution :

a.json

```
 {
   "employee": [
      {
        "id": "01",
        "name": "Amit",
        "department": "Sales"
      },
      {
        "id": "04",
        "name": "sunil",
        "department": "HR"
      }
   ]
}
```

----

a.py

```
import json

# Opening JSON file
f = open('a.json',)

# returns JSON object as
# a dictionary
data = json.load(f)

# Iterating through the json
# list
for i in data['employee']:
    print(i)

# Closing file
f.close()
```

c)Persisting JSON

```
import json
import pymongo

# Create a JSON file
data = {"name": "John Doe", "age": 30}
with open("a.json", "w") as f:
   json.dump(data, f)

# Persist the JSON file in a database
client = pymongo.MongoClient("localhost", 27017)
db = client.test # db name
collection = db.json # collection name

# Create a document from the JSON file
document = json.load(open("a.json"))
collection.insert_one(document)
print("Data inserted successfully")
# Close the database connection
client.close()
```