

IMPORT A JSON FILE FROM THE COMMAND LINE.

APPLY THE FOLLOWING ACTIONS WITH THE DATA,

PRESENT IN THE JSON FILE WHERE PROJECTION, AGGREGATION, REMOVE, COUNT, SKIP & SORT.

Step 1: Create a json file named "data.json" using the command

\$nano data.json

Enter some data in the json file.

```
[  
  { "name": "Alice", "age": 25, "city": "New York" },  
  { "name": "Bob", "age": 30, "city": "Los Angeles" },  
  { "name": "Charlie", "age": 35, "city": "Chicago" },  
  { "name": "David", "age": 40, "city": "Houston" },  
]
```

Now, while projecting the data, we enter the command `$jq '[.] | {name,age}]' data.json` to display a specific values.

```
dell@dell-Inspiron-3443:~$ nano expt6.json  
dell@dell-Inspiron-3443:~$ jq '[.] | {name,age}]' data.json  
jq: error: Could not open file data.json: No such file or directory  
dell@dell-Inspiron-3443:~$ jq '[.] | {name,age}]' expt6.json  
[  
  {  
    "name": "Alice",  
    "age": 25  
  },  
  {  
    "name": "Bob",  
    "age": 30  
  },  
  {  
    "name": "Charlie",  
    "age": 35  
  },  
  {  
    "name": "Dogo",  
    "age": 29  
  }  
]
```

Step 2: Using the command `$jq '[.[] | .age] | add / length' data.json` we get the aggregate value of the data.

Step 3: Using the command `$jq 'map(del(.city))' data.json` we remove some specified values of the data.

```
dell@dell-Inspiron-3443:~$ jq '[.[] | .age] | add / length' expt6.json
29.75
dell@dell-Inspiron-3443:~$ jq 'map(del(.city))' expt6.json
[
  {
    "name": "Alice",
    "age": 25
  },
  {
    "name": "Bob",
    "age": 30
  },
  {
    "name": "Charlie",
    "age": 35
  }
]
```

Step 4: Using the command `$jq 'length' data.json` we get the total count of the data.

Step 5: Using the command `$jq '[0:2]' data.json` we assign the limit of the data to be displayed.

```
dell@dell-Inspiron-3443:~$ jq '[0:2]' expt6.json
[
  {
    "name": "Alice",
    "age": 25,
    "city": "New York"
  },
  {
    "name": "Bob",
    "age": 30,
    "city": "Los Angeles"
  }
]
dell@dell-Inspiron-3443:~$ jq 'length' expt6.json
4
```

Step 6: Using the command `$jq '[2:]' data.json`, it skips some of the data and display the remaining data.

```
dell@dell-Inspiron-3443:~$ jq '[0:2]' expt6.json
[
  {
    "name": "Alice",
    "age": 25,
    "city": "New York"
  },
  {
    "name": "Bob",
    "age": 30,
    "city": "Los Angeles"
  }
]
```

```
dell@dell-Inspiron-3443:~$ jq '[2:]' expt6.json
[
  {
    "name": "Charlie",
    "age": 35,
    "city": "Chicago"
  },
  {
    "name": "Dogo",
    "age": 29,
    "city": "Carolina"
  }
]
```

Step 7: Using the command `$jq 'sort_by(.age)' data.json`, it sorts the data in a specified criteria.

```
dell@dell-Inspiron-3443:~$ jq 'sort_by(.age)' expt6.json
[
  {
    "name": "Alice",
    "age": 25,
    "city": "New York"
  },
  {
    "name": "Dogo",
    "age": 29,
    "city": "Carolina"
  },
  {
    "name": "Bob",
    "age": 30,
    "city": "Los Angeles"
  },
  {
    "name": "Charlie",
    "age": 35,
    "city": "Chicago"
  }
]
```