* MongoDB stores data in the form of *documents*, which are JSON-like **field and value pairs**.
* Documents are analogous to structures in programming languages that associate keys with values (e.g. dictionaries, hashes, maps, and associative arrays).
* MongoDB documents are *BSON* documents. BSON is a binary representation of JSON with additional type information.
* In the documents, the value of a field can be any of the BSON data types, including other documents, arrays, and arrays of documents.
* MongoDB stores douments in collections. Collections are analogous to a table in relational databases.
* A write operation is any operation that creates or modifies data in MongoDB instance.
* In MongoDB, a write operation targets a single collection.
* All write operations in MongoDB are atomic on the level of a single document.
* 3 types of write operations.
  + *Insert :* Adds new documents to the collection.
  + *Update:* Modifies existing documents.
  + *Delete:* Deletes documents from a collection.
* No insert, update or delete can affect more than one document atomically.
* For the update and remove operations, you can specify criteria, or filters, that identify the documents to update or remove. These operations use the same query syntax to specify the criteria as read operations.
* The *\_id* field is required in every MongoDB document. It acts as a primary key of the document.
* If you add a new document without the *\_id* field, the client library or the mongod instance adds an *\_id* field and populates the field with a unique ObjectId.
* If you pass in an *\_id* value that already exists, an exception is thrown. The *\_id* field is uniquely indexed by default in every collection