```
> now we going to update
now we going to update
    ^^

Uncaught SyntaxError: Unexpected identifier
> Movie.updateOne({title:dune}, {year: 2021}).then(res => console.log(res, res.n, res.modifi…
Uncaught ReferenceError: dune is not defined
> Movie.updateOne({title:"dune"}, {year: 2021}).then(res => console.log(res, res.n, res.modi…
Promise {
  <pending>,
  [Symbol(async_id_symbol)]: 4689,
  [Symbol(trigger_async_id_symbol)]: 4685
}
> {
  acknowledged: true,
  modifiedCount: 1,
  upsertedId: null,
  upsertedCount: 0,
  matchedCount: 1
} undefined undefined
> Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10), function(err,data){ conso…
Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10), function(err,data){ console…
                                                               ^


Uncaught SyntaxError: Unexpected token ')'
> Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10} function(err, data){consol…
Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10} function(err, data){console.…
                                                               ^


Uncaught SyntaxError: missing ) after argument list
> Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10}, function(err, data){conso…
Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10}, function(err, data){console…
                                                                                             …

Uncaught SyntaxError: missing ) after argument list
> Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10} function(err, data){consol…
Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10} function(err, data){console.…
                                                               ^


Uncaught SyntaxError: missing ) after argument list
> Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10}, function(err, data){conso…
Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10}, function(err, data){console…
                                                                                             …

Uncaught SyntaxError: missing ) after argument list
> Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10}, function(err, data){conso…
Query {
  _mongooseOptions: {},
  _transforms: [],
  _hooks: Kareem { _pres: Map(0) {}, _posts: Map(0) {} },
  _executionStack: null,
  mongooseCollection: Collection {
    collection: Collection { s: [Object] },
    Promise: [Function: Promise],
    modelName: 'Movie',
    _closed: false,
    opts: {
      autoIndex: true,
      autoCreate: true,
      schemaUserProvidedOptions: {},
      capped: false,
      Promise: undefined,
      '$wasForceClosed': undefined
    },
    name: 'movies',
    collectionName: 'movies',
    conn: NativeConnection {
      base: [Mongoose],
      collections: [Object],
      models: [Object],
      config: {},
      replica: false,
      options: null,
      otherDbs: [],
      relatedDbs: {},
      states: [Object: null prototype],
      _readyState: 1,
      _closeCalled: undefined,
      _hasOpened: true,
      plugins: [],
      id: 0,
      _queue: [],
      _listening: false,
      _connectionOptions: [Object],
      _connectionString: 'mongodb://127.0.0.1:27017/test',
      client: [MongoClient],
      '$initialConnection': [Promise],
      db: [Db],
      host: '127.0.0.1',
      port: 27017,
      name: 'test'
    },
    queue: [],
    buffer: false,
    emitter: EventEmitter {
      _events: [Object: null prototype] {},
      _eventsCount: 0,
      _maxListeners: undefined,
      [Symbol(kCapture)]: false
    }
  },
  model: Model { Movie },
  schema: Schema {
    obj: {
      title: [Function: String],
      year: [Function: Number],
      rating: [Function: Number],
      isGood: [Function: Boolean]
    },
    paths: {
      title: [SchemaString],
      year: [SchemaNumber],
      rating: [SchemaNumber],
      isGood: [SchemaBoolean],
      _id: [ObjectId],
      __v: [SchemaNumber]
    },
    aliases: {},
    subpaths: {},
    virtuals: { id: [VirtualType] },
    singleNestedPaths: {},
    nested: {},
    inherits: {},
    callQueue: [],
    _indexes: [],
    methods: {},
    methodOptions: {},
    statics: {},
    tree: {
      title: [Function: String],
      year: [Function: Number],
      rating: [Function: Number],
      isGood: [Function: Boolean],
      _id: [Object],
      __v: [Function: Number],
      id: [VirtualType]
    },
    query: {},
    childSchemas: [],
    plugins: [ [Object], [Object], [Object], [Object], [Object] ],
    '$id': 1,
    mapPaths: [],
    s: { hooks: [Kareem] },
    _userProvidedOptions: {},
    options: {
      typeKey: 'type',
      id: true,
      _id: true,
      validateBeforeSave: true,
      read: null,
      shardKey: null,
      discriminatorKey: '__t',
      autoIndex: null,
      minimize: true,
      optimisticConcurrency: false,
      versionKey: '__v',
      capped: false,
      bufferCommands: true,
      strictQuery: false,
      strict: true,
      pluralization: true
    },
    '$globalPluginsApplied': true,
    _requiredpaths: []
  },
  op: 'updateMany',
  options: {},
  _conditions: { title: { '$in': [Array] } },
  _fields: undefined,
  _updateDoc: undefined,
  _path: undefined,
  _distinctDoc: undefined,
  _collection: NodeCollection {
    collection: Collection {
      collection: [Collection],
      Promise: [Function: Promise],
      modelName: 'Movie',
      _closed: false,
      opts: [Object],
      name: 'movies',
      collectionName: 'movies',
      conn: [NativeConnection],
      queue: [],
      buffer: false,
      emitter: [EventEmitter]
    },
    collectionName: 'movies'
  },
  _traceFunction: undefined,
  '$useProjection': true,
  _update: { rating: 10 }
}
> Uncaught MongooseError: Query.prototype.exec() no longer accepts a callback
    at model.Query.exec (/home/ironic/Desktop/Frontend-Web/MongooseBasics/node_modules/mongo…
    at _update (/home/ironic/Desktop/Frontend-Web/MongooseBasics/node_modules/mongoose/lib/q…
    at Query.updateMany (/home/ironic/Desktop/Frontend-Web/MongooseBasics/node_modules/mongo…
    at _update (/home/ironic/Desktop/Frontend-Web/MongooseBasics/node_modules/mongoose/lib/m…
    at Function.updateMany (/home/ironic/Desktop/Frontend-Web/MongooseBasics/node_modules/mo…
> Movie.updateMany({ title: {$in: ["dune", "dark"]}}, {rating:10} ).then(data => console.log…
Promise {
  <pending>,
  [Symbol(async_id_symbol)]: 7592,
  [Symbol(trigger_async_id_symbol)]: 7588
}
> {
  acknowledged: true,
  modifiedCount: 2,
  upsertedId: null,
  upsertedCount: 0,
  matchedCount: 2
}
> Movie.findOneAndUpdate({{year: 2021} , {title: "Dune"} ).then(data => console.log(data))
Movie.findOneAndUpdate({{year: 2021} , {title: "Dune"} ).then(data => console.log(data))
                        ^

Uncaught SyntaxError: Unexpected token '{'
> Movie.findOneAndUpdate({year: 2021} , {title: "Dune"} ).then(data => console.log(data))
Promise {
  <pending>,
  [Symbol(async_id_symbol)]: 8776,
  [Symbol(trigger_async_id_symbol)]: 8772
}
> {
  _id: new ObjectId("644ddf84abb3044f0e3f9cab"),
  title: 'dune',
  year: 2021,
  rating: 10,
  isGood: true,
  __v: 0
}
> there we got the old version of the updated one on console. Actually the data is updated. …
there we got the old version of the updated one on console. Actually the data is updated. it…
      ^^

Uncaught SyntaxError: Unexpected identifier
> Movie.findOneAndUpdate({year: 2021} , {rating: 9.6}, {new: true} ).then(data => console.lo…
Promise {
  <pending>,
  [Symbol(async_id_symbol)]: 9969,
  [Symbol(trigger_async_id_symbol)]: 9965
}
> {
  _id: new ObjectId("644ddf84abb3044f0e3f9cab"),
  title: 'Dune',
  year: 2021,
  rating: 9.6,
  isGood: true,
  __v: 0
}
> note: we used the {new: true} in previous statement
note: we used the {new: true} in previous statement
         ^^^^

Uncaught SyntaxError: Unexpected identifier
>
```