

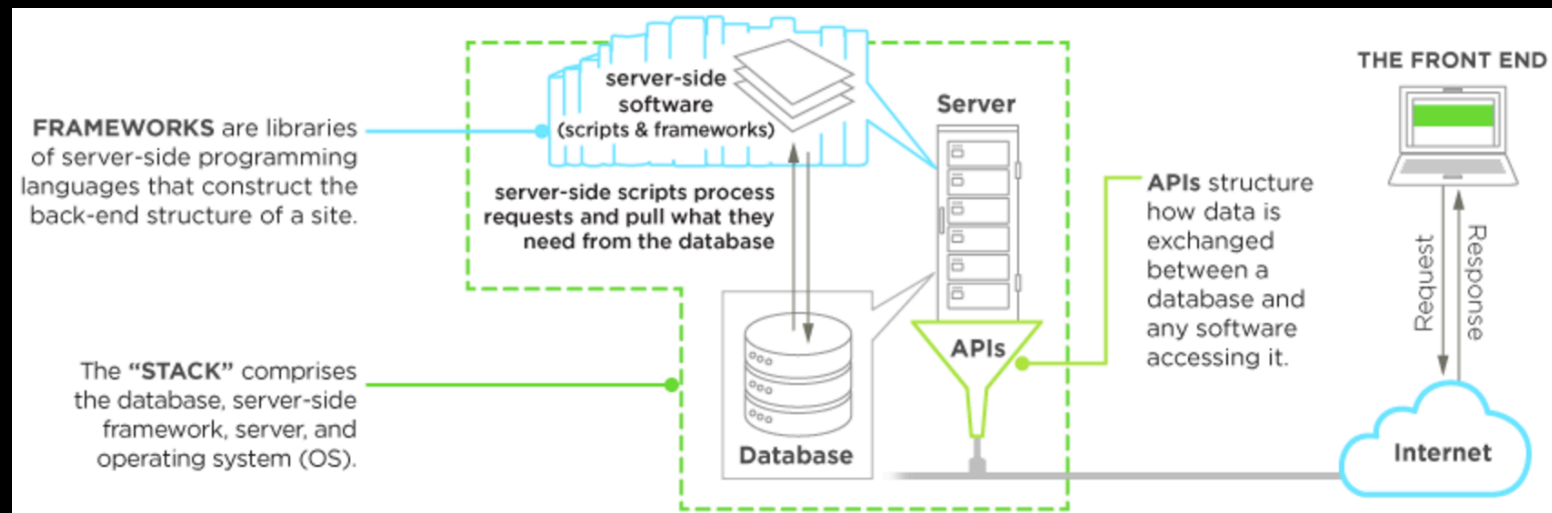
# Web Development

Node js

# Server-Side Scripting

Back-End Web Development Technology

## A Quick look at your website's back end



## The back end comprises three parts

- ★ The server, your database, any APIs
- ★ API(Application Program Interface): a set of routines, protocols, and tools for building software applications.
- ★ APIs written by server-side languages: Java, PHP, Python, Ruby, C#, C++ ...

# Server-Side Scripting

## Server-side script basics

- ★ Runs on a server, embedded in the site's code
- ★ Designed to interact with back-end permanent storage, like databases
- ★ Facilitates the transfer of data from server to browser
- ★ Runs on-call. When a webpage is “called up,”, server-side scripts process and return data
- ★ Powers functions in dynamic web applications, such as user validation, saving and retrieving data, and navigating between other pages
- ★ Build application programming interfaces (APIs), which control what data and software a site shares with other apps

## OPEN APIs

- ★ OAuth (Open Authorization), Facebook API ...
- ★ Mashup services: Maps, Search engines, Shopping products..

# Node.js

## What is Node.js

- ★ Node.js is a server-side platform built on Google Chrome's JavaScript Engine
- ★ Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications.

## Features of Node.js

- ★ **Asynchronous and Event driven:** non-blocking.
- ★ **Very fast:** Node.js library is very fast in code execution
- ★ **Single Threaded** but Highly Scalable: Node.js uses a single threaded model with event looping.
- ★ **No Buffering:** Node.js applications never buffer any data.

## What Can Node.js Do

- ★ Web application
- ★ Mobile application, desktop application

# Node.js Get Started

## Download & Install

File download from <https://nodejs.org/> and run it.

## Your first Node.js - Hello world

Create a file named "helloworld.js" in your workspace, and add the following code

```
var http = require('http');  
  
http.createServer(function (req, res) {  
  res.end('Hello World!');  
}).listen(8080);
```

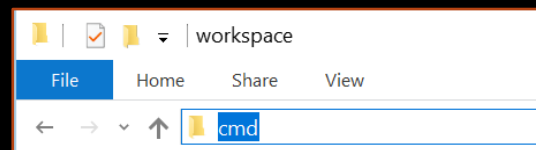
## Command Line Interface

- ★ Node.js files must be initiated in the "Command Line Interface" program
- ★ Run: Press the start button and look for "Command Prompt", or simply write "cmd" in the search field.

# Node.js Get Started

## Open “Command Prompt”

- ★ Open “File explorer” and go to your workspace folder
- ★ Type “cmd” in the search field.

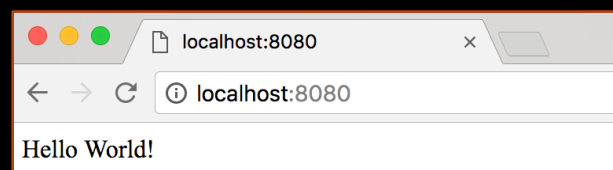


## Run helloworld.js

- ★ Type “node helloworld.js” in your Command Prompt

```
D:\workspace\YourName> node helloworld.js
```

- ★ Start your web browser, and type in the address: <http://localhost:8080>



## Stop Node.js

- ★ Press “Ctrl + C” in your Command Prompt.

# Node.js Modules

## What is a Module in Node.js

Consider modules to be the same as JavaScript libraries.

## Built-in Modules

A set of built-in modules which you can use without any further installation.

## Include Modules

To include a module, use the `require()` function with the name of the module

```
var http = require('http');
```

Now your application has access to the HTTP module

```
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/plain'});  
  res.end('Hello World!');  
}) .listen(8080);
```

# Node.js Modules

## Create Your Own Modules

Create a new file named “mymodule.js”, and add following code.

```
exports.myDateTime = function () {  
  return Date();  
};
```

exports: declare new module function

Add following code in “helloworld.js”

```
var http = require('http');  
var dt = require('./mymodule');  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write("Now : " + dt.myDateTime() + "<br>");  
  res.end('Hello World!');  
}).listen(8080);
```

Declare my module

Content-type: text/plain

Type “node helloworld” in your Command Prompt

Confirm on web: <http://127.0.0.1:8080>

## Mission:

Add now function add(a,b) in “mymodule.js” and use it



# HTTP Modules

## The Built-in HTTP Module

allows Node.js to transfer data over the Hyper Text Transfer Protocol (HTTP)

## Node.js as a Web Server

- ★ can create an HTTP server that listens to server ports and gives a response back to the client
- ★ Create a file named “demo.js”, and add following code.

```
var http = require('http');  
  
http.createServer(function (req, res) {  
  
    res.writeHead(200, {'Content-Type': 'text/html'});  
    res.write('Hello World!');  
    res.end();  
}).listen(80);  
console.log('Server is running...');
```

Include http module : require()

req, res: request, response object

write(): write a response data

writeHead(): add an http header

end(): end the response

The server listen on port 80

logging

## Run & Confirm on web (http://localhost)

```
D:\workspace\YourName> node demo
```

# URL Modules

## Read & Parse the Query String

- ★ req argument that represents the request from the client, as an object
- ★ Add following code in "demo.js"

```
var url = require('url');
http.createServer(function (req, res) {
  var r = url.parse(req.url, true);
  console.log('request url:', req.url);
  console.log('url parse', r);
  ...
}).listen(80);
```

Include url module : require()

req.url: request url  
url.parse: parse url

## Run & Confirm on web

<http://localhost/search?key=pet&page=1>

# Sample server: index page

Create a file named “server.js”, and add following code.

```
var http = require('http');  
var url = require('url');  
  
http.createServer(function (req, res) {  
  var reqUrl = url.parse(req.url, true);  
  var html = ' ';  
  switch(reqUrl.pathname){  
    case '/':  
      html = '<h1> Index page </h1>';  
      html += '<a href="/useradd.html">Add user</a><br>';  
      html += '<a href="/userlist.html">User list</a>';  
      break;  
    default:  
      res.writeHead(404, {'Content-Type': 'text/html'});  
      res.end('Action not found: ' + reqUrl.pathname);  
      break;  
  }  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.end(html);  
}).listen(8080);  
console.log('Server is running...');
```

Include module: http, url

.pathname : request path

http://localhost:8080

# Sample server: /useradd.html

Add page: /useradd.html

Add following code in "server.js"

```
switch(reqUrl.pathname){  
  ...  
  case '/useradd.html':  
    html = '<h1> User add </h1>';  
    html += '<form action="/useradd.do">';  
    html += '  User name: <input type="text" name="uname"><br>';  
    html += '  User email: <input type="email" name="uemail"><br>';  
    html += '  <button onclick="go(this.form);">Add user</button> ';  
    html += '</form>';  
    html += '<script>';  
    html += '  console.log("page loaded... ' + new Date()+' ');';  
    html += '  function go(f){';  
    html += '    alert("User name: " + f.uname.value);';  
    html += '    f.submit();';  
    html += '  }';  
    html += '</script>';  
    break;
```

# Sample server: /useradd.do

## Add action: /useradd.do

Add following code in "server.js"

```
switch(reqUrl.pathname){  
    ...  
    case '/useradd.do':  
        html = '<h1> Action called : /useradd.do </h1>';  
        html += 'User name: ' + reqUrl.query.uname + '<br>';  
        html += 'User email: ' + reqUrl.query.uemail + '<br>';  
        break;
```

# Sample server: /userlist.html

## Add: /userlist.html, userlist.do

Add following code in "server.js"

```
switch(reqUrl.pathname){  
  ...  
  case '/userlist.html':  
    html = '<h1> User list </h1>';  
    html += '<form action="/userlist.do">';  
    html += '<input type="search" name="key">';  
    html += '<input type="submit" value="Search">';  
    html += '</form>';  
    break;  
  
  case '/userlist.do':  
    html = '<h1> Action called: /userlist.do </h1>';  
    html += ' Search key: ' + reqUrl.query.key;  
    break;  
}
```

# Sample server: Make html file

## Create html files

### index.html

```
<h1> Index page </h1>
<a href="/useradd.html">Add user</a><br>
<a href="/userlist.html">User list</a>
```

### useradd.html

```
<h1> User add </h1>
<form action="/useradd.do">
  User name: <input type="text" name="uname"><br>
  User email: <input type="email" name="email"><br>
  <input type="submit" value="Add User">
</form>
```

### userlist.html

```
<h1> User list </h1>
<form action="/userlist.do">
  <input type="search" name="key">
  <input type="submit" value="Search">
</form>
```

# Sample server: Read html file

**Add following code in "server.js"**

```
var fs = require('fs');

http.createServer(function (req, res) {
  ...
  if(reqUrl.pathname.endsWith('.do')){
    switch(reqUrl.pathname){
      case '/useradd.do':
        ...
      case '/userlist.do':
        ...
      default:
    }
    res.writeHead(status, {'Content-Type': 'text/html'});
    res.end(html);
  } else {
  }
}
```

Include module: fs

String.endsWith()

Remove path  
/useradd.html  
/userlist.html

Add code here: Nexxe page



# Sample server: Read file

Add following code in "server.js"

```
} else {  
  var filename = '.' + reqUrl.pathname;  
  if(filename == './') filename = './index.html';  
  fs.readFile(filename, function(err,data){  
    if(err) {  
      res.writeHead(404, {'Content-Type': 'text/html'});  
      res.end('File not found: '+ reqUrl.pathname);  
    }  
    var ctype = 'text/html';  
    if(reqUrl.pathname.startsWith('/js/')){  
      ctype = 'text/javascript';  
    } else if(reqUrl.pathname.startsWith('/css/')){  
      ctype = 'text/css';  
    } else if(reqUrl.pathname.startsWith('/images/')){  
      ctype = 'images/' + /^[^.] +$/ .exec(filename);  
    }  
    res.writeHead(200, {'Content-Type': ctype});  
    res.end(data);  
  });  
}
```

Current folder

Contents type  
String.startsWith()

File file extension

# Sample server: Refactoring

## Make action module

Create a new file named “user.js”, and add following code

```
var url = require('url');
```

```
exports.add = function(req, res){  
  var q = url.parse(req.url, true).query;  
  var html = '<h1> Action called: useradd.do </h1>';  
  html += ' User name: ' + q.uname + '<br>';  
  html += ' User email: ' + q.uemail + '<br>';  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.end(html);  
};
```

Include url module : require()

add function

```
exports.list = function(req, res){  
  var q = url.parse(req.url, true).query;  
  var html = '<h1> Action called: userlist.do </h1>';  
  html += ' Search key: ' + q.key;  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.end(html);  
};
```

list function

# Sample server: Refactoring

## Modify “server.js”

```
var user = require('./user');
```

Include custom module

```
http.createServer(function (req, res) {
```

```
  ...
```

```
  case '/useradd.do':
```

```
    user.add(req, res);
```

```
    break;
```

```
  case '/userlist.do':
```

```
    user.list(req, res);
```

```
    break;
```

Call custom module

# NPM

## What is NPM?

- ★ NPM is a **package manager** for Node.js packages, or modules if you like.
- ★ NPM is already ready to run on your computer!

## Download a Package

- ★ Command prompt> npm install "package-name"
- ★ Ex. Install MongoDB package (in your workspace folder)

```
D:\workspace\YourName> npm install mongodb
```

## Using a Package

```
var mongo = require('mongodb');
```

## Install Nodemon

Node.js Automatic re-running (for developing): nodemon filename.js

```
D:\workspace\YourName> npm install -g nodemon
```

-g: Install to global

# MongoDB

## Install MongoDB

Download & Install : <https://www.mongodb.com/>

## Run Mongodb

- ★ Make database folder: d:\workspace\db
- ★ Command prompt> mongod --dbpath "database path"

```
D:\workspace\YourName> mongod --dbpath d:\workspace\db
```

## Install node.js package

Command prompt> npm install mongodb

```
D:\workspace\YourName> npm install mongodb
```

# MongoDB

## Creating Database

Create a file named “demo\_create\_mongo\_db”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/mydb";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  console.log("Database created!");
  db.close();
});
```

## Run in terminal

```
D:\workspace\YourName> node demo_create_mongo_db.js
```

# MongoDB

## Creating Collection

Create a file named “demo\_mongodb\_createcollection”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  //Create a collection name "customers":
  dbo.createCollection("customers", function(err, res) {
    if (err) throw err;
    console.log("Collection created!");
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_mongodb\_createcollection.js

# MongoDB: Insert Data

## Inserting Data

Create a file named “demo\_mongodb\_insert”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myobj = { name: "Company Inc", address: "Highway 37" };
  dbo.collection("customers").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
```

## Run in terminal

```
D:\workspace\YourName> node demo_mongodb_insert.js
```



# MongoDB: Insert Data

## Inserting Data

Create a file named “demo\_mongodb\_insert\_multiple”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myobj = [{ name: 'John', address: 'Highway 71'},
    { name: 'Peter', address: 'Lowstreet 4'},
    { name: 'Amy', address: 'Apple st 652'},
    { name: 'Hannah', address: 'Mountain 21'},
    { name: 'Michael', address: 'Valley 345'},
    { name: 'Sandy', address: 'Ocean blvd 2'},
    { name: 'Betty', address: 'Green Grass 1'},
    { name: 'Richard', address: 'Sky st 331'},
    { name: 'Susan', address: 'One way 98'},
    { name: 'Vicky', address: 'Yellow Garden 2'},
    { name: 'Ben', address: 'Park Lane 38'},
    { name: 'William', address: 'Central st 954'},
    { name: 'Chuck', address: 'Main Road 989'},
    { name: 'Viola', address: 'Sideway 1633'}];
  dbo.collection("customers").insertMany(myobj, function(err, res) {
    if (err) throw err;
    console.log("Number of documents inserted: " + res.insertedCount);
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_mongodb\_insert\_multiple.js

# MongoDB: Insert Data

## Inserting Data

Create a file named “demo\_mongodb\_insert\_id”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myobj = [
    { _id: 154, name: 'Chocolate Heaven' },
    { _id: 155, name: 'Tasty Lemon' },
    { _id: 156, name: 'Vanilla Dream' }
  ];
  dbo.collection("products").insertMany(myobj, function(err, res) {
    if (err) throw err;
    console.log(res);
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_mongodb\_insert\_id.js

# MongoDB: Find Data

## Finding Data

Create a file named “demo\_mongodb\_findone”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  dbo.collection("customers").findOne({}, function(err, result) {
    if (err) throw err;
    console.log(result.name);
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_mongodb\_findone.js

# MongoDB: Find Data

## Finding Data

Create a file named “demo\_mongodb\_find”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  dbo.collection("customers").find({}).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_mongodb\_find.js

# MongoDB: Find Data

## Finding Data

Create a file named “demo\_mongodb\_find\_fields”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  dbo.collection("customers").find({}, { projection: { _id: 0, name: 1, address: 1 }
}).toArray(function(err, result) {
  if (err) throw err;
  console.log(result);
  db.close();
});
});
```

## Run in terminal

D:\workspace\YourName> node demo\_mongodb\_find\_fields.js

# MongoDB: Query

## Finding Data With Query

Create a file named “demo\_mongodb\_query”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var query = { address: "Park Lane 38" };
  dbo.collection("customers").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_mongodb\_query.js

# MongoDB: Sorting

## Sorting Data

Create a file named “demo\_sort”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var mysort = { name: 1 };
  dbo.collection("customers").find().sort(mysort).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

```
{name: 1 } // ascending
```

```
{ name: -1 } // descending
```

## Run in terminal

D:\workspace\YourName> node demo\_sort.js

# MongoDB: Delete

## Deleting Data

Create a file named “demo\_delete”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myquery = { address: 'Mountain 21' };
  dbo.collection("customers").deleteOne(myquery, function(err, obj) {
    if (err) throw err;
    console.log("1 document deleted");
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_delete.js



# MongoDB: Delete

## Deleting Data

Create a file named “demo\_delete\_many”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myquery = { address: /^0/ };
  dbo.collection("customers").deleteMany(myquery, function(err, obj) {
    if (err) throw err;
    console.log(obj.result.n + " document(s) deleted");
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_delete\_many.js

# MongoDB: Drop

## Drop Collection

Create a file named “demo\_dropcollection”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  dbo.collection("products").drop(function(err, obj) {
    if (err) throw err;
    console.log(" collection dropped");
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_dropcollection.js

# MongoDB: Update

## Updating Data

Create a file named “demo\_update\_one”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://127.0.0.1:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myquery = { address: "Valley 345" };
  var newvalues = { $set: {name: "Mickey", address: "Canyon 123" } };
  dbo.collection("customers").updateOne(myquery, newvalues, function(err, res) {
    if (err) throw err;
    console.log("1 document updated");
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_update\_one.js

# MongoDB: Update

## Updating Data

Create a file named “demo\_update\_many”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://127.0.0.1:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myquery = { address: /^S/ };
  var newvalues = {$set: {name: "Minnie"} };
  dbo.collection("customers").updateMany(myquery, newvalues, function(err, res) {
    if (err) throw err;
    console.log(res.result.nModified + " document(s) updated");
    db.close();
  });
});
```

## Run in terminal

D:\workspace\YourName> node demo\_update\_many.js

# MongoDB: Limit

## Limiting Data

Create a file named “demo\_mongodb\_limit”, and add following code.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  dbo.collection("customers").find().limit(5).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
})
```

## Run in terminal

D:\workspace\YourName> node demo\_mongodb\_limit.js

**Mission Complete**