### Node.js Modules

* A module is an encapsulated and reusable chunk of code that has its own context
* In Node.js each file is treated as a separate module
* A Module is same as JavaScript libraries.
* A Module is a set of functions that can use in application

### Type of Modules

* Local modules- Modules that we create in our application
* Built-in modules- Modules that Node.js ships with out of the box
* Third party modules- Modules written by other developers that we can use in our application

### Built-in Modules

* Node.js has a set of built-in modules which you can use without any further installation
* Ex: http,fs etc.
* To include a module, use the require() function with the name of the module:
  + var http = require('http')
  + var  fs=require('fs')

### Create Your Own Modules(Local Modules)

* You can create your own modules, and easily include them in your applications.

The following example creates a module that returns a date and time object:

Example

Create a module that returns the current date and time:

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

Use the exports keyword to make properties and methods available outside the module file.

Save the code above in a file called "myfirstmodule.js"

Include Your Own Module

Now you can include and use the module in any of your Node.js files.

Example

Use the module "myfirstmodule" in a Node.js file:

var http = require('http');  
**var dt = require('./myfirstmodule');**  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write("The date and time are currently: " + **dt.myDateTime()**);  
  res.end();  
}).listen(8080);

[Run example »](https://www.w3schools.com/nodejs/shownodejs.asp?filename=demo_module)

Notice that we use ./ to locate the module, that means that the module is located in the same folder as the Node.js file.

Save the code above in a file called "demo\_module.js", and initiate the file:

Initiate demo\_module.js:

C:\Users\*Your Name*>node demo\_module.js

If you have followed the same steps on your computer, you will see the same result as the example: [http://localhost:8080](http://localhost:8080/)