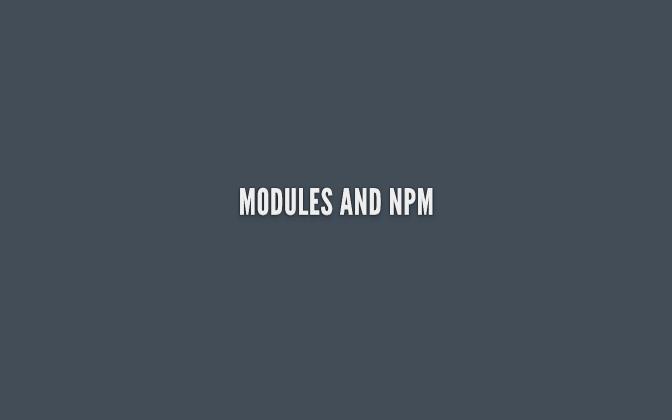
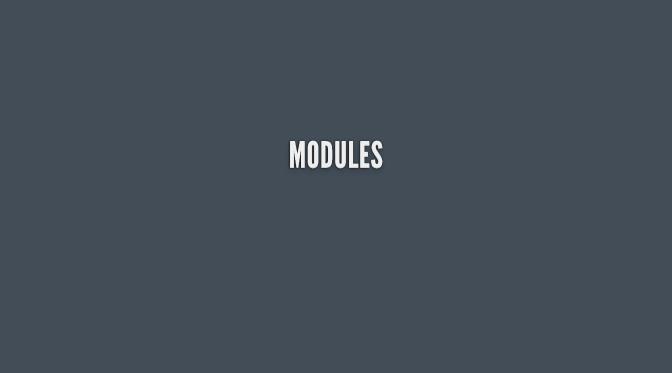
# NODE FIRM

Converbt® 2013 The Node Firm All Dights December





## **ANATOMY OF A MODULE**

A module is a JavaScript file that exports an object.

A module is a JavaScript file.

A module is a file.

#### Example:

```
module.exports = function getPI() {
  return Math.PI;
};
```

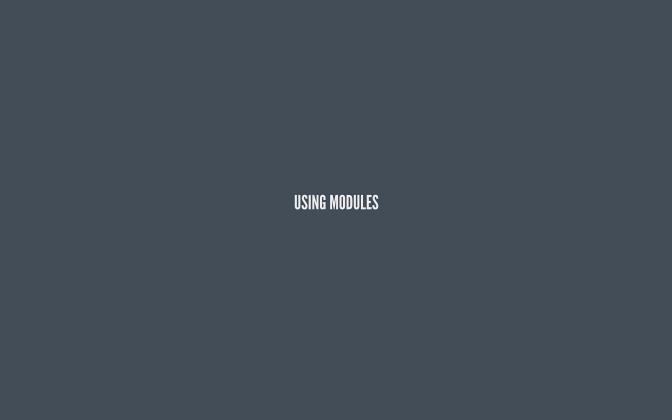
#### A unit of logic such as:

```
var pi = Math.PI;
function diameter(radius) {
  return2 * radius;
}
function circumference(radius) {
  return pi * diameter(radius);
}
function area(radius) {
  return pi * Math.pow(radius, 2);
}
```

## **CAN BE A MODULE:**

# ...with private code **01\_circle.js:**

```
var pi = Math.PI;
module.exports = function(radius) {
  function diameter() {
    return 2 * radius;
  }
  function circumference() {
    return pi * diameter();
  }
  function area() {
    return pi * Math.pow(radius, 2);
  }
  return {
    area: area,
    circumference: circumference
  };
}
```



#### FROM THE REPL

#### You can also omit the .js:

```
> var Circle = require('./01_circle');
```

#### DE 00110ED110E0D0

JAVASCRIPT CONSTRUCTORS

are functions!

Uppercased Circle informs the reader of the intent of the function.

### REQUIRE require() is a qualified import:

- Does not modify the environment (returns the module exported value)
  Resolves to a full file path, not conflicting with any other module

## OTHER GLOBALS

- \_\_filename path to the current file
- \_\_dirname path to the current file's directory

## **USER MODULES**

require can be called with a file path (absolute or relative to the current file):

require('./lib/shapes/circle.js');
require('../util.js');

## **CORE MODULES**

require can be called with the name of a core module:

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

There are several core modules like http, net, assert, dgram, path, url, util and others.

## **MODULE CACHING**

```
$ node
> var request = require('request');
> request.hello = "world";
> console.log(require('request').hello);
world
undefined
```

Modules are only loaded from disk once



## COMMAND LINE NPM

a Unix-like tool

- \$ npm search request
  \$ npm doc request
- \$ npm install request
- \$ npm remove request

## **INSTALLING MODULES**

npm is bundled with Node.js.
Install the request module:

\$ npm install request

#### NPM installs in a local node\_modules folder:

#### Back in the REPL:

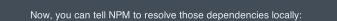
- \$ node
  > war request = require('request');
  - Module search paths for module names (not a file path)
    - core module
       node\_modules folder in current directory
    - 3. node modules folder in parent directories

#### **PACKAGE.JSON**

the package manifest

#### package.json:

```
"name": "modules-src",
"version": "0.0.1",
"description": "Modules unit source code",
"dependencies": {
    "request": "*",
    "async": "*",
}
}
```



- this will ensure all the dependencies in your package.json are installed
- \$ node

- \$ npm install



Modules **separate** program logic into discrete, manageable units.