

NODE.JS MODULES

Stephen Schaub

Node.js Module System

2

- Node.js standard library organized into modules
- A package is one or more modules grouped for distribution
- Package Manager: npm
 - ▣ Installs packages hosted at [npmjs.org](https://www.npmjs.org)

Package Locations

3

- Packages can be installed locally or globally
- Local install:
 - ▣ `npm install packagename`
 - ▣ Installs module into `node_modules` subdirectory of current working directory
- Global install:
 - ▣ `npm install -g packagename`
 - ▣ Installs module into shared location
 - Windows: In `user_profile_directory\AppData\Roaming\npm`
 - ▣ Packages containing command line tools should be installed globally so their tools will be in the PATH

Installed Packages

4

- Display installed packages:

- ▣ npm ls

- Displays Local packages

- ▣ npm ls -g

- Displays Global packages

- Update packages:

- ▣ npm update *packagename* [-g]

- Uninstall packages:

- ▣ npm rm *packagename* [-g]

Using Modules in Programs

5

- Use `require()` to import code from installed modules

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

- Search order:
 1. Core modules compiled into Node.js
 2. `node_modules` folder in current directory
 3. `node_modules` folder in parent directories

Using Modules in Programs

6

- `require()` can specify path to module

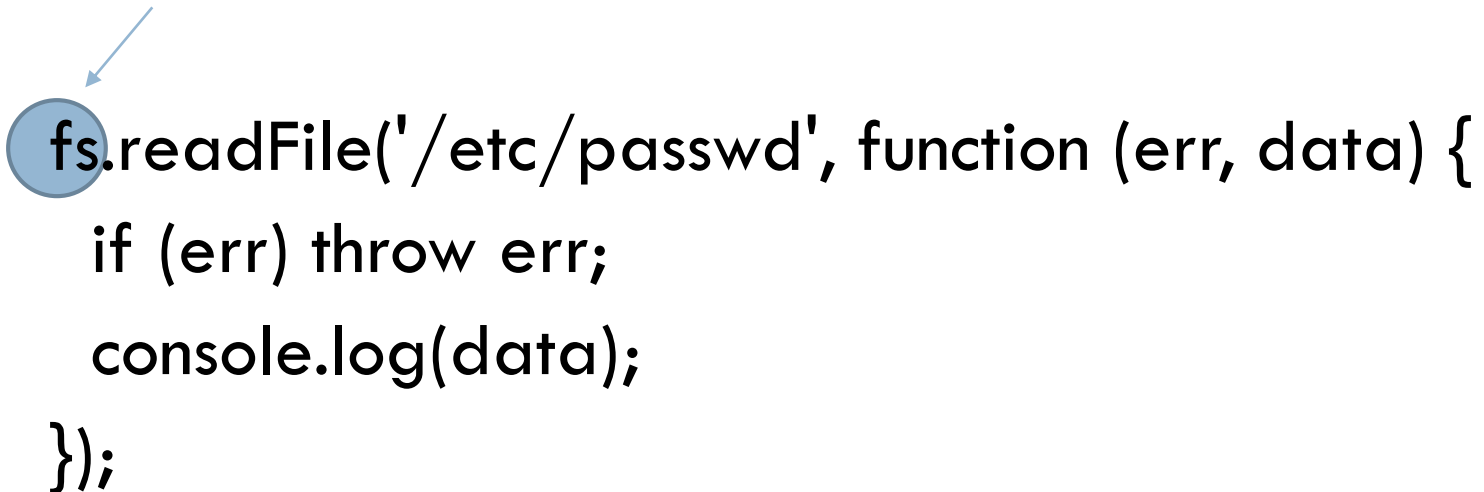
```
var mymodule = require('./mymodule');
```

- ▣ Looks for module in `./mymodule` (rather than in `./node_modules/mymodule`)

Using a Module

7

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



```
fs.readFile('/etc/passwd', function (err, data) {  
    if (err) throw err;  
    console.log(data);  
});
```

- The object returned from `require()` is used to invoke functions defined in the module

8

Creating Modules

Module Structure

9

- A Node.js module is a folder that contains a collection of JavaScript files and other resource files
- A config file named `package.json` specifies info about the module
- Minimal example:

```
{  
  "name": "package-name",  
  "version": "0.0.0"  
}
```

package.json

10

- Module entry point:
 - ▣ "main": "index.js"
 - ▣ Specifies the file to be loaded when module is requested with `require()`
 - ▣ Default is `index.js`

index.js

11

- May define variables and functions
- Variables and functions are private to the module
- To export a function, assign it to a property of a predefined variable named **exports**
 - ▣ In scripts that `require()` the module, **exports** is returned from `require()`

Module Example

12

- `./mymodule/index.js`

```
exports.hello = function() {  
    console.log("You called me!");  
}
```

- `./demo.js`

```
mymodule = require("./mymodule");  
mymodule.hello();  
// invokes hello in mymodule
```

Module Example

13

- See `examples/nodejs/module_demo`