

# Node.js

Dominic Bosch

March 26, 2014

## What is it

General Attributes

## The callback paradigm

Program and Forget  
setTimeout Example

## Everything is a Module in Node.js

Built-in Modules  
Node Packaged Modules (NPM)  
Custom JS Modules

## ► Software Platform

- ▶ Software Platform
- ▶ Scalable Networking applications

- ▶ Software Platform
- ▶ Scalable Networking applications
- ▶ Asynchronous events, Non-blocking I/O

- ▶ Software Platform
- ▶ Scalable Networking applications
- ▶ Asynchronous events, Non-blocking I/O
- ▶ Applications written in JavaScript

- ▶ Software Platform
- ▶ Scalable Networking applications
- ▶ Asynchronous events, Non-blocking I/O
- ▶ Applications written in JavaScript
- ▶ Many built-in libraries

- ▶ Software Platform
- ▶ Scalable Networking applications
- ▶ Asynchronous events, Non-blocking I/O
- ▶ Applications written in JavaScript
- ▶ Many built-in libraries
- ▶ Custom libraries from Package Manager NPM



## Synchronous functions:

```
var fs = require('fs');  
try {  
  var data = fs.readFileSync('./cryptico.js', 'utf8');  
} catch (err) {  
  
}
```

## Synchronous functions:

```
var fs = require('fs');  
try {  
  var data = fs.readFileSync('./cryptico.js', 'utf8');  
} catch (err) {  
  
}
```

## Asynchronous functions:

```
var fs = require('fs');  
var fAsyncCalled = function(err, data) {  
  if(err){  
    console.log('Error during file retrieval!');  
    return;  
  }  
};  
fs.readFile('./cryptico.js', 'utf8', fAsyncCalled);
```

## Why is this not going to work?

```
for( var i = 0; i < 100; i++ ) {  
  setTimeout( function() {  
    console.log(i);  
  }, 1000);  
}
```

## Solution:

```
var fDelayed = function(id) {  
  return function() {  
    console.log(id);  
  };  
};  
for( var i = 0; i < 100; i++ ) {  
  setTimeout( fDelayed(i), 1000);  
}
```

Registers the value of `i` by value and not by reference in the anonymous function which is returned on the first function call of `fDelayed(i)`.

## (Almost) Real-life example:

```
function fetchUserObjectsAsync( arrNames, cbAnswer ){
  var semaphore = arrNames.length,
      objUsers = {},
      fGetUserAsync = function( name, cbLocal ){
        db.getUser(name, cbLocal);
      },
      fJoin = function( uName ) {
        return function( err, obj ) {
          --semaphore
          if( err || !obj ) return;
          objUsers[ uName ] = obj;
          if( semaphore === 0 ){
            cbAnswer( null, objReplies );
          }
        }
      };

  for( var name in arrNames ){
    fGetUserAsync( user, fJoin( name ) );
  }
}
```

- ▶ Load Modules via require:

- ▶ Load Modules via require:
  - ▶ Built-in: `require('fs');`

- ▶ Load Modules via require:
  - ▶ Built-in: `require('fs');`
  - ▶ Project: `require('./my-module');`



- ▶ Load Modules via require:
  - ▶ Built-in: `require('fs');`
  - ▶ Project: `require('./my-module');`
- ▶ Modules are cached!

<http://nodejs.org/api/>

- ▶ Url: <https://www.npmjs.org/>

- ▶ Url: <https://www.npmjs.org/>
- ▶ Everybody can publish to NPM!

- ▶ Url: <https://www.npmjs.org/>
- ▶ Everybody can publish to NPM!
- ▶ Global/Local/Project installation

My Favorite: Project Installation

package.json

npm install

track version

Example of how people still could break your roject: `groc/jade`

place index.js into node-modules - cryptico.js