How To Write
Node.js Module

Fred Chien

Warning

警告！

今天會很悶

!!!!!!!!!!!!!!!!!!!!!!!!

我盡量講的

!!!!!!!!!!!!!!!!!!!!!!!!

WHO AM I ?

我是誰？

Fred Chien
錢逢祥

永遠的大四生

Fred Chien
錢逢祥

Mandice CEO

fred-zone.blogspot.com

cfsghost@gmail.com

people.linux.org.tw/~fred/

cfsghost@gmail.com

Topics.

Node.js Modules

NPM Registry

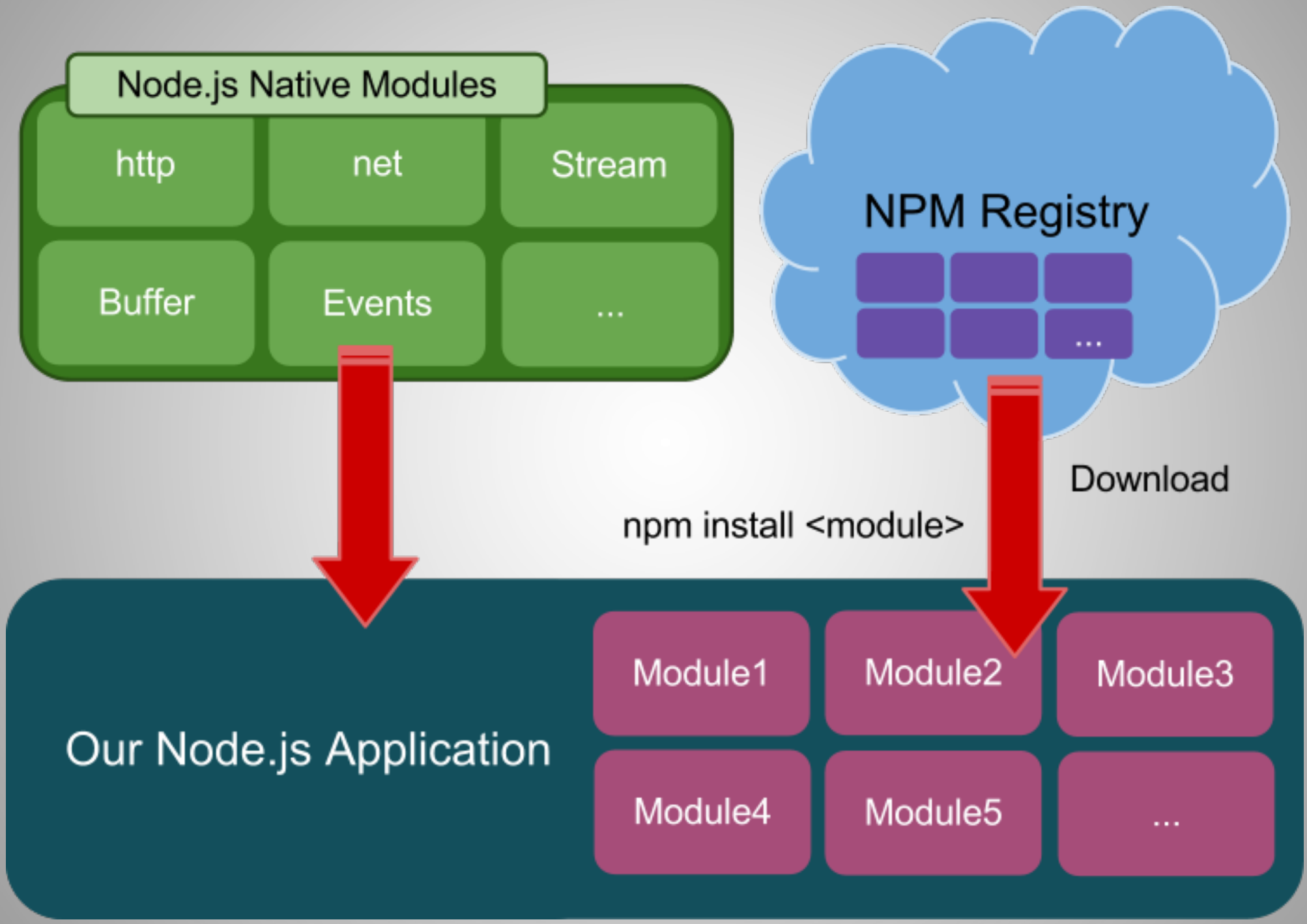
C/C++ Addons

What is
Node.js Module ?

Node.js 模組?

npm install <*something*>

You Must Be Familiar With


Node.js Native Modules
http
net
Stream
Buffer
Events
...
NPM Registry
...
Download
npm install <module>
Our Node.js Application
Module1
Module2
Module3
Module4
Module5
...

You Can Write Module In C/C++ & JavaScript

你可以用 C/C++ 或 JavaScript 寫模組

How to
Load Module

載入模組

Load Global Module

Example:

var MyModule = require('mymodule');

Node.js will searching in the following location:

- *./node_modules*
- *../node_modules*
- *$HOME/.node_modules*
- *$HOME/.node_libraries*
- *$PREFIX/lib/node*

Load Local Module

Load the module in the same directory:

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

Or

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

Write The First
Node.js Module

動手寫第一個模組

The First Module Example

```
module.exports = function() {

    console.log('Hello World!');

};
```

require() ⟺ module.exports

Bridge between app and module

Implement a Class in Module

```
module.exports = function() {
    var self = this;
    this.counter = 0;

    this.pump = function() {
        self.counter++;
    };

};
```

More JavaScript Styles

```
var Pumper = module.exports = function() {
    this.counter = 0;
};

Pumper.prototype.pump = function() {
    Pumper.counter++;
};
```

Export Objects and Constants

```
var Pumper = module.exports.Pumper = function() {
    this.counter = 0;
};

Pumper.prototype.pump = function() {
    Pumper.counter++;
};

module.exports.Pumper1 = function() { ... };
module.exports.Pumper2 = function() { ... };
module.exports.Birthday = 714;
```

index.js
&
index.node

./example/index.js

```
var ex = require('./example');
```

如果是目錄，預設讀取 index.js 或 index.node

Let's
Publish Your Module

釋出！與你的朋友分享成果吧！

NPM Registry

NPM = Node Package Manager

NPM Registry

npmjs.org

Steps to Publish Package

打包並上傳模組的步驟

1. Get **NPM Account**
2. Generate **package.json**
3. To **Upload** Package

Get NPM Account

註冊 NPM 帳號

npm adduser

新增 NPM 帳號

Initialize Package

初始化你的套件

npm init

產生 package. json

Run "npm init"

```
$ npm init
Package name: (demo)
Description: Hello
Package version: (0.0.0)
Project homepage: (none)
Project git repository: (none)
Author name: Fred
Author email: (none) cfsghost@gmail.com
Author url: (none)
Main module/entry point: (none)
Test command: (none)
```

We got package.json

```
{
  "author": "Fred <cfsghost@gmail.com>",
  "name": "demo",
  "description": "Hello",
  "version": "0.0.0",
  "repository": {
    "url": ""
  },
  "dependencies": {},
  "devDependencies": {},
  "optionalDependencies": {},
  "engines": {
    "node": "*"
  }
}
```

Normal Structure of Package

- index.js
- package.json
- README (README.md)
- LICENSE
- lib/hello1.js
- lib/hello2.js
- tests/test1.js
- tests/test2.js

I don't want to use index.js !

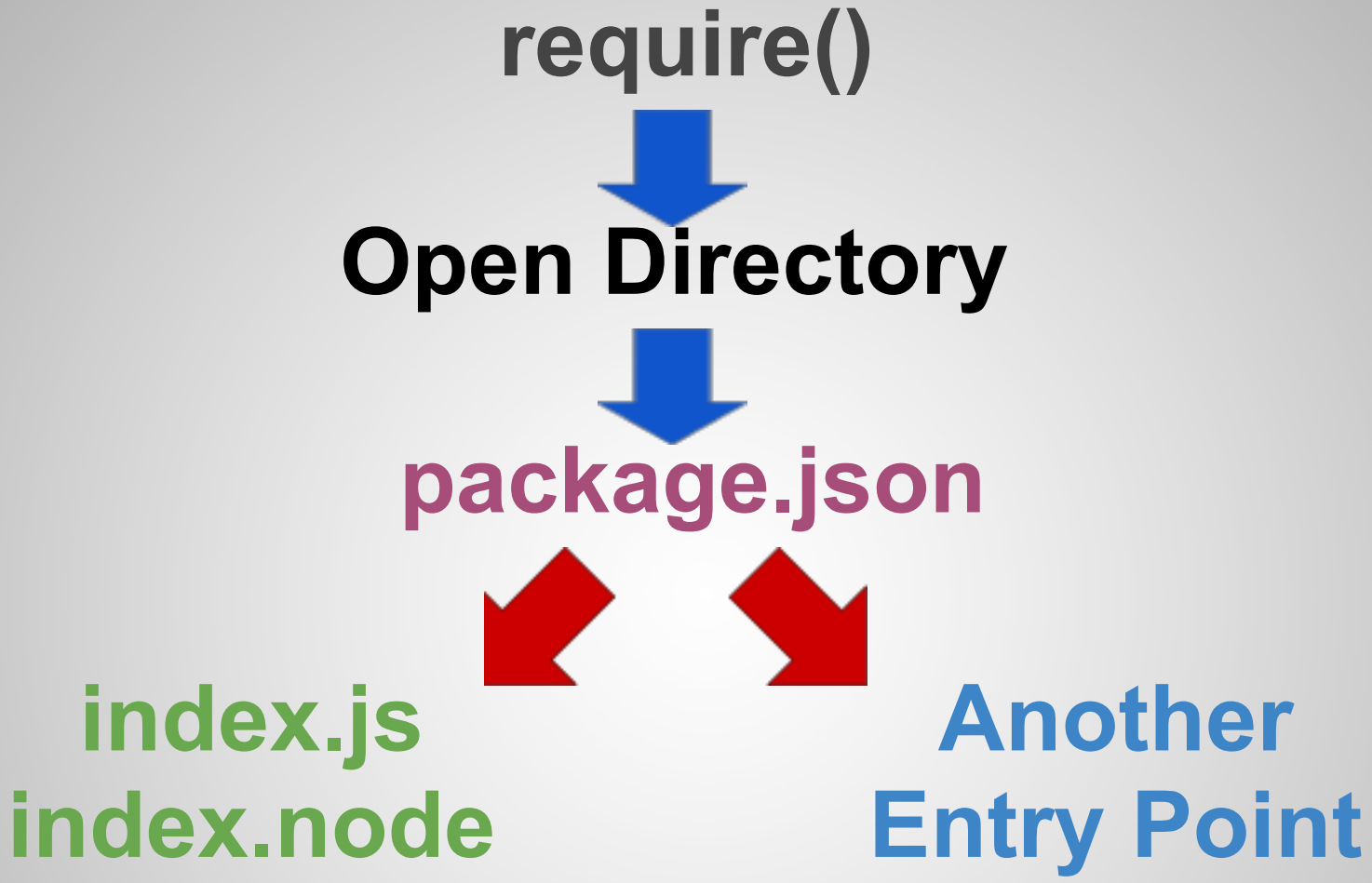
Change Entry Point

我想改變進入啟始點

Add "main" Property To package.json

After Change Entry Point

- demo.js
- package.json
- README (README.md)
- LICENSE
- lib/hello1.js
- lib/hello2.js
- tests/test1.js
- tests/test2.js


require()
Open Directory
package.json
index.js
index.node
Another
Entry Point

Upload Package

上傳套件！公開於世！

npm publish .

發佈在 '.' 之下的 Package

Piece of cake!

開發模組一點都不難嘛！

進階

Advanced Topic

How to
Write C/C++ Addons

如何使用 C/C++ 寫模組

Development Environment

1. **GCC** (used to compile)
2. **Python** (For build script)

Write The First
C/C++ Addon

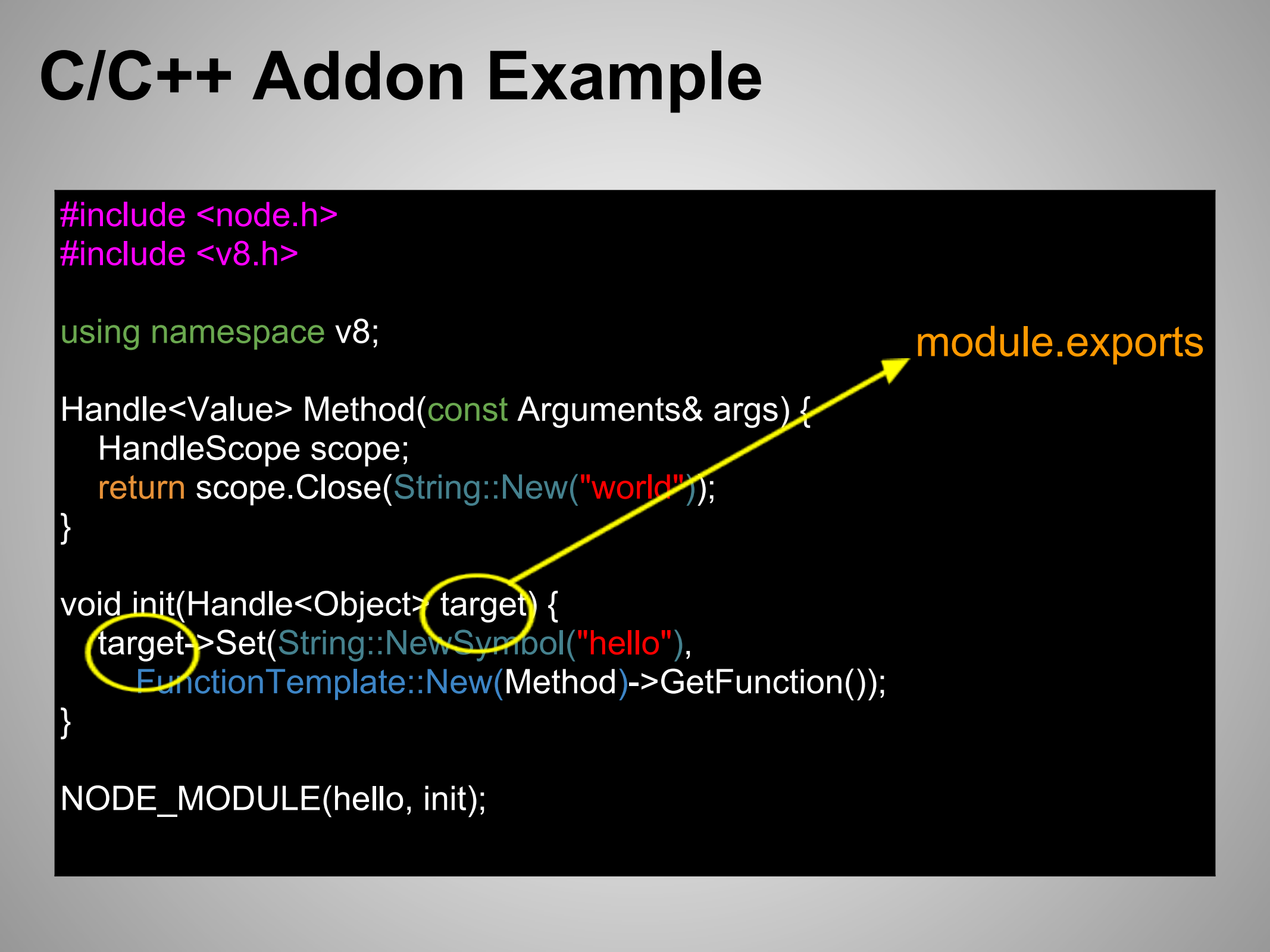
動手寫第一個 C/C++ 模組

C/C++ Addon Example

```
#include <node.h>
#include <v8.h>

using namespace v8;

Handle<Value> Method(const Arguments& args) {
    HandleScope scope;
    return scope.Close(String::New("world"));
}

void init(Handle<Object> target) {
    target->Set(String::NewSymbol("hello"),
        FunctionTemplate::New(Method)->GetFunction());
}

NODE_MODULE(hello, init);
```

C/C++ Addon Example

```
#include <node.h>
#include <v8.h>

using namespace v8;

Handle<Value> Method(const Arguments& args) {
    HandleScope scope;
    return scope.Close(String::New("world"));
}

void init(Handle<Object> target) {
    target->Set(String::NewSymbol("hello"),
        FunctionTemplate::New(Method)->GetFunction());
}

NODE_MODULE(hello, init);
```



C/C++ Addon Example

```
#include <node.h>
#include <v8.h>

using namespace v8;

Handle<Value> Method(const Arguments& args) {
    HandleScope scope;
    return scope.Close(String::New("world"));
}

void init(Handle<Object> target) {
    target->Set(String::NewSymbol("hello"),
        FunctionTemplate::New(Method)->GetFunction());
}

NODE_MODULE(hello, init);
```

```
function() {
    return 'world';
}
```

Compare with JavaScript Version

```
var target = module.exports;

target['hello'] = function() {
    return 'world!';
};
```

Or

```
module.exports.hello = function() {
    return 'world!';
};
```

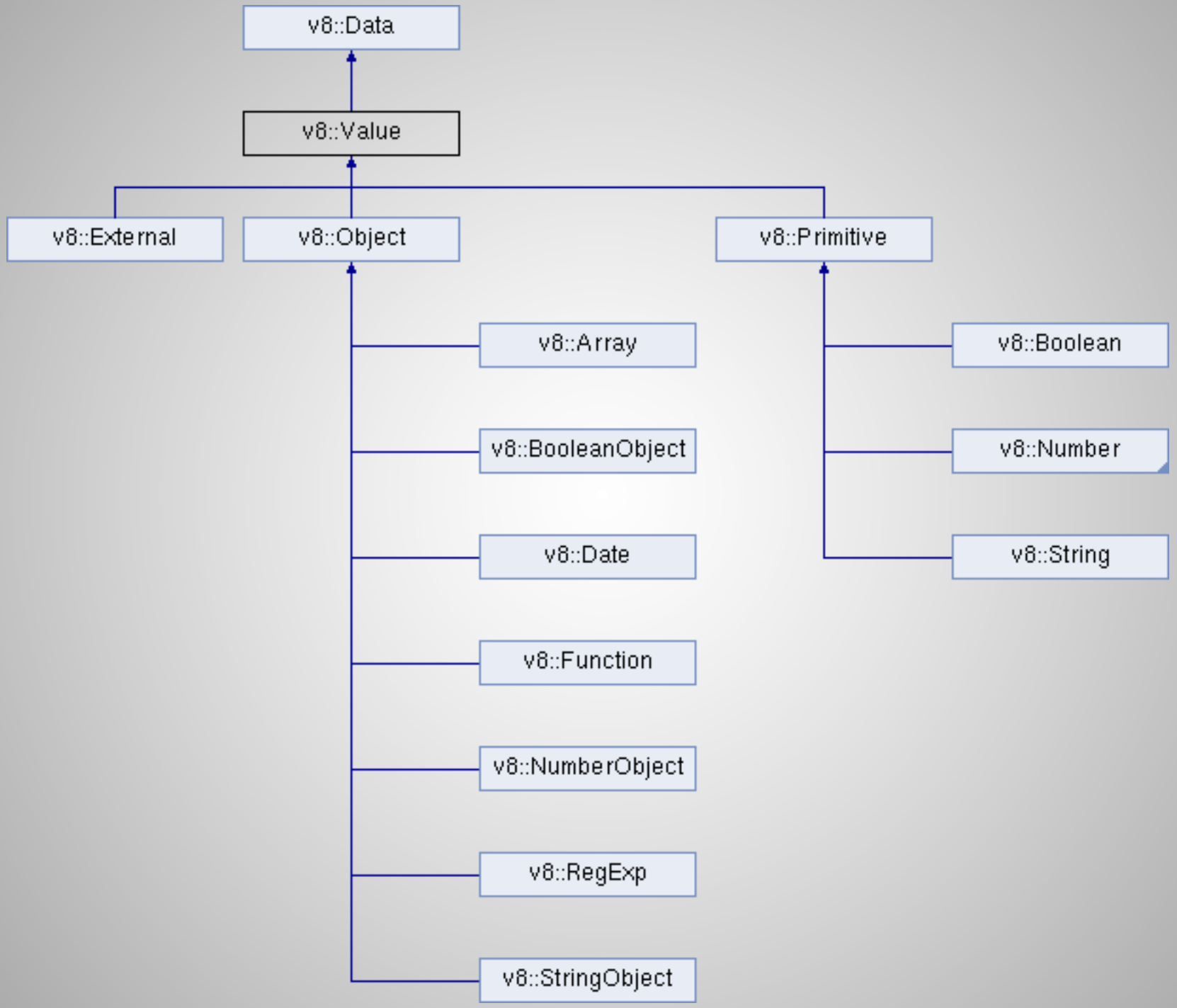
C/C++ Addon Example

```
#include <node.h>
#include <v8.h>

using namespace v8;

Handle<Value> Method(const Arguments& args) {
    HandleScope scope;
    return scope.Close(String::New("world"))
}

void init(Handle<Object> target) {
    target->Set(String::NewSymbol("hello"),
        FunctionTemplate::New(Method)->GetFunction());
}

NODE_MODULE(hello, init);
```

C/C++ Addon Example

```
#include <node.h>
#include <v8.h>

using namespace v8;

Handle<Value> Method(const Arguments& args) {
    HandleScope scope;
    return scope.Close(String::New("world"));
}

void init(Handle<Object> target) {
    target->Set(String::NewSymbol("hello"),
        FunctionTemplate::New(Method)->GetFunction());
}

NODE_MODULE(hello, init);
```

v8::String Class


v8::Data
v8::Value
v8::External
v8::Object
v8::Primitive
v8::Array
v8::BooleanObject
v8::Date
v8::Function
v8::NumberObject
v8::RegExp
v8::StringObject
v8::Boolean
v8::Number
v8::String

C/C++ Addon Example

```
#include <node.h>
#include <v8.h>

using namespace v8;

Handle<Value> Method(const Arguments& args) {
    HandleScope scope;
    return scope.Close(String::New("world"))
}

void init(Handle<Object> target) {
    target->Set(String::NewSymbol("hello"),
        FunctionTemplate::New(Method)->GetFunction());
}

NODE_MODULE(hello, init);
```

HandleScope

- Determine Lifetime of handles
- Often created at the beginning of a function call
- Deleted after function return
  - **scope.Close(***<Handle>***)** to avoid handle being deleted by **Garbage Collection**

Compile The First
C/C++ Addon

動手編譯第一個 C/C++ 模組

You Must Have

wscript

你必需有一個 wscript

wscript

```
srcdir = '.'
blddir = 'build'
VERSION = '0.0.1'

def set_options(opt):
  opt.tool_options('compiler_cxx')

def configure(conf):
  conf.check_tool('compiler_cxx')
  conf.check_tool('node_addon')

def build(bld):
  obj = bld.new_task_gen('cxx', 'shlib', 'node_addon')
  obj.target = 'hello'
  obj.source = 'hello.cc'
```

Structure of Package

- package.json
- README (README.md)
- LICENSE
- wscript
- hello.cc

node-waf configure build

使用 node-waf 編譯我們的程式！

Generated
build/Release/hello.node

編譯產生 Binary 檔案

Write A Test Case

```
var Hello = require('./build/Release/hello.node');

console.log(Hello.hello());
```

Don't Forget This Before Upload Package

上傳模組前要修改 package.json

Add "**scripts**" Property To **package.json**

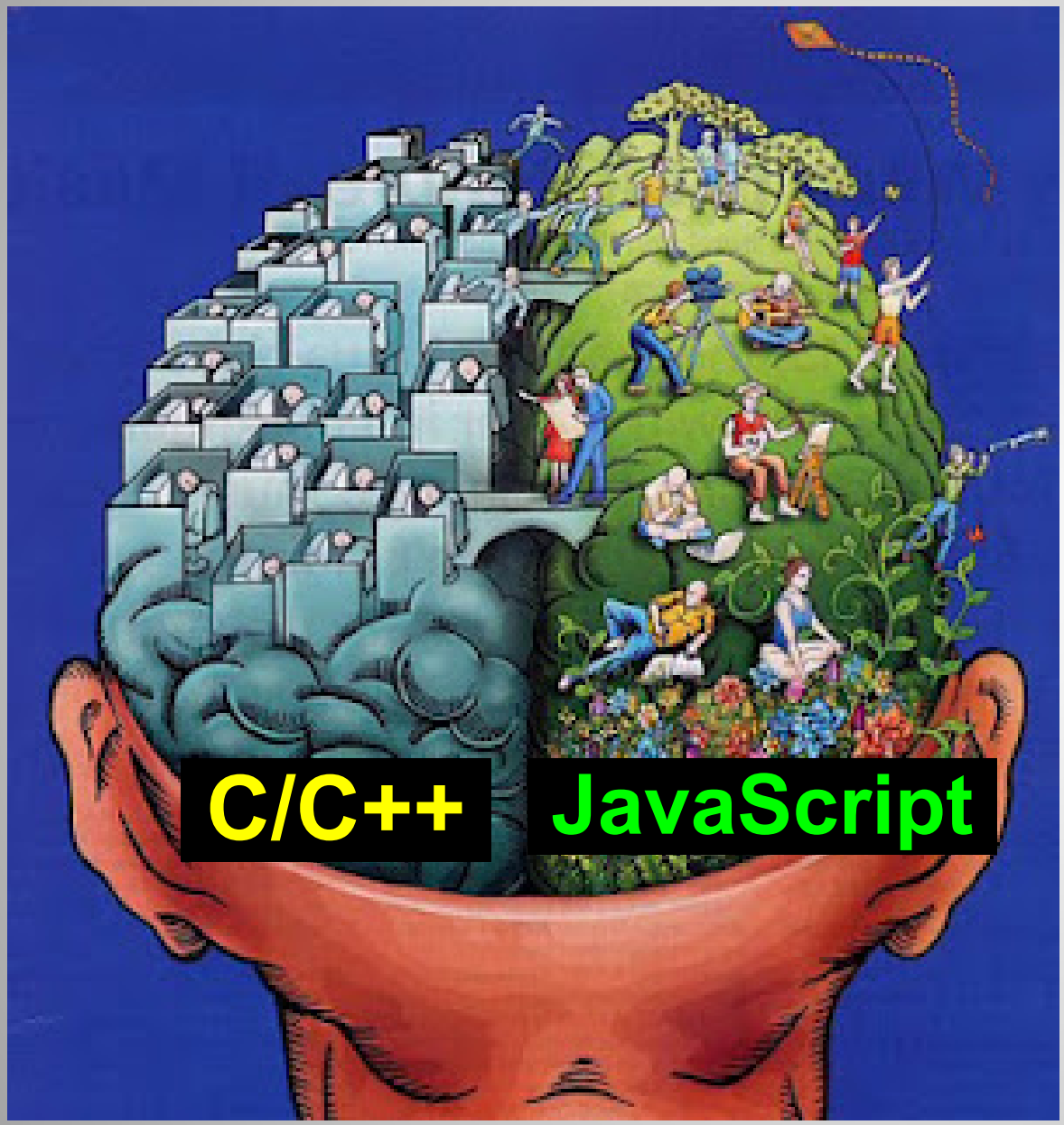
Modify package.json

```
{
    "name": "hello",
    ...
    "main": "build/Release/hello.node",
    "scripts": {
        "install": "node-waf configure build"
    }
}
```

npm publish .

發佈我們的 C/C++ Addon！

開發 C/C++ Addon 的竅門



人腦內建：

1. C/C++ Compiler
2. JavaScript Compiler
3. Virtual Machine

進階 Part 2

Advanced Topic Part 2

Arguments

參數

Assume We Have Parameters

```
var Hello = require('./build/Release/hello.node');

console.log(Hello.hello('String', 101, 4.0, true));
```

Get Arguments In C/C++

```
Handle<Value> Method(const Arguments& args) {
    HandleScope scope;

    printf("%d\n", args.Length());

    if (args[0]->IsString() && args[1]->IsNumber() &&
        args[2]->IsNumber() && args[3]->IsBoolean()) {

        printf("%s %d %f %d\n",
            *String::AsciiValue(args[0]->ToString()),
            args[1]->ToInteger()->Value(),
            args[2]->NumberValue(),
            args[3]->ToBoolean());
    }

    return scope.Close(String::New("world"))
}
```

如果以上能理解

進一步討論 C/C++ 與 JavaScript 的關係

Understanding of Types

了解 JavaScript 資料型態

Types In JavaScript

```
var string = 'Hello String';
var integer = 714;
var float = 11.15;
var boolean = false;
var obj = {};
var arr = [];
var func = function() {};
```

Methods

** Assume Local<Value> data;*

Type Name	Check Type	Get Value
String	IsString()	*String::Utf8Value(data->ToString()) *String::AsciiValue(data->ToString())
Integer	IsNumber() IsInt32() IsUint32()	data->ToInteger()->Value() data->ToInt32()->Value() data->ToUint32()->Value() data->IntegerValue() data->Uint32Value() data->Int32Value()
Float	IsNumber()	data->NumbeValue()
Boolean	IsBoolean()	data->ToBoolean()->Value()
Object/Array	IsObject()	data->ToObject()

Function	IsFunction()	Local<Function> cb = Local<Function>::Cast(data);
		const unsigned argc = 1;
		Local<Value> argv[argc] = {
		Local<Value>::New(String::New("hello world"))
		};
		cb->Call(Context::GetCurrent()->Global(), argc, argv);

Class

參數

Create Class Instance

```
var Hello = require('./build/Release/hello.node');

var hello = new Hello.Hello();

console.log(hello.method());
```

Class Constructor in JavaScript

```
var Hello = function() {
    /* Constructor */
};
module.exports.Hello = Hello;

/* Prototype Method */
Hello.prototype.myMethod = function() {
    return 'World';
};
```

Write a Constructor

```
#include <node.h>
#include <v8.h>

using namespace v8;

Handle<Value> MyConstructor(const Arguments& args) {
    HandleScope scope;

    return args.This();
}

void init(Handle<Object> target) {
    Local<FunctionTemplate> tpl = FunctionTemplate::New(MyConstructor);

    target->Set(String::NewSymbol("Hello"),
      tpl->GetFunction());
}

NODE_MODULE(hello, init);
```

Write a Prototype Method

```
...

Handle<Value> MyMethod(const Arguments& args) {
    HandleScope scope;
   return scope.Close(String::New("World"));
}

void init(Handle<Object> target) {
    Local<FunctionTemplate> tpl = FunctionTemplate::New(MyConstructor);

    tpl->InstanceTemplate()->SetInternalFieldCount(1);
    NODE_SET_PROTOTYPE_METHOD(tpl, "myMethod", MyMethod);

    target->Set(String::NewSymbol("Hello"),
       tpl->GetFunction());
}

NODE_MODULE(hello, init);
```

一切只是開端

It's beginning to do

進階 Part 3

To Be Continued...

這真的是進階

I am NOT Kidding You

Wrapping C++ Object

將 C++ Class 包裝成 JavaScript Class

Create JavaScipt Instance In C/C++

Do "new Hello.Hello()" in C/C++

See You Next Time

下次有機會見面再來討論吧！

喔！差點忘了

火力展示

Question?

歡迎發問

Thanks

感